
CCCCCCCCCCC OOOOOOOOO NNN NNN VVV VVV
CCCCCCCCCCC OOOOOOOOO NNN NNN VVV VVV
CCCCCCCCCCC OOOOOOOOO NNN NNN VVV VVV
CCC OOO OOO NNN NNN VVV VVV
CCC OOO OOO NNN NNN VVV VVV
CCC OOO OOO NNN NNN VVV VVV
CCC OOO OOO NNNNNN NNN VVV VVV
CCC OOO OOO NNNNNN NNN VVV VVV
CCC OOO OOO NNNNNN NNN VVV VVV
CCC OOO OOO NNN NNN NNN VVV VVV
CCC OOO OOO NNN NNN NNN VVV VVV
CCC OOO OOO NNN NNN NNN VVV VVV
CCC OOO OOO NNN NNNNNN VVV VVV
CCC OOO OOO NNN NNNNNN VVV VVV
CCC OOO OOO NNN NNNNNN VVV VVV
CCC OOO OOO NNN NNN VVV VVV
CCC OOO OOO NNN NNN VVV VVV
CCC OOO OOO NNN NNN VVV VVV
CCCCCCCCCCC OOOOOOOOO NNN NNN VVV
CCCCCCCCCCC OOOOOOOOO NNN NNN VVV
CCCCCCCCCCC OOOOOOOOO NNN NNN VVV

FILEID**CONVFILES

```
  CCCCCCCC    000000    NN      NN  VV      VV  FFFFFFFFFF    IIIIII    LL          EEEEEEEEEE    SSSSSSSS
  CCCCCCCC    000000    NN      NN  VV      VV  FFFFFFFFFF    IIIIII    LL          EEEEEEEEEE    SSSSSSSS
CC          OO      OO  NN      NN  VV      VV  FF              II      LL          EE          SS
CC          OO      OO  NN      NN  VV      VV  FF              II      LL          EE          SS
CC          OO      OO  NNNN    NN  VV      VV  FF              II      LL          EE          SS
CC          OO      OO  NNNN    NN  VV      VV  FF              II      LL          EE          SS
CC          OO      OO  NN  NN  NN  VV      VV  FFFFFFFF        II      LL          EEEEEEEE      SSSSSS
CC          OO      OO  NN  NN  NN  VV      VV  FFFFFFFF        II      LL          EEEEEEEE      SSSSSS
CC          OO      OO  NN    NNNN  VV      VV  FF              II      LL          EE                  SS
CC          OO      OO  NN    NNNN  VV      VV  FF              II      LL          EE                  SS
CC          OO      OO  NN      NN    VV  VV    FF              II      LL          EE                  SS      ....
CC          OO      OO  NN      NN    VV  VV    FF              II      LL          EE                  SS      ....
  CCCCCCCC    000000    NN      NN      VV      FF            IIIIII    LLLLLLLLLL  EEEEEEEEEE  SSSSSSSS      ....
  CCCCCCCC    000000    NN      NN      VV      FF            IIIIII    LLLLLLLLLL  EEEEEEEEEE  SSSSSSSS      ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
;  1          0001  0 %TITLE  'VAX-11 CONVERT'
;  2          0002  0 MODULE  CONV$FILES      ( IDENT='V04-000',
;  3          0003  0                         OPTLEVEL=3
;  4          0004  0                         ) =
;  5          0005  0
;  6          0006  1 BEGIN
;  7          0007  1
;  8          0008  1 !**************************************************************************
;  9          0009  1 !*                                                                        *
; 10          0010  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                             *
; 11          0011  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.               *
; 12          0012  1 !*  ALL RIGHTS RESERVED.                                                 *
; 13          0013  1 !*                                                                        *
; 14          0014  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
; 15          0015  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE *
; 16          0016  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER *
; 17          0017  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
; 18          0018  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY *
; 19          0019  1 !*  TRANSFERRED.                                                           *
; 20          0020  1 !*                                                                        *
; 21          0021  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE *
; 22          0022  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT *
; 23          0023  1 !*  CORPORATION.                                                           *
; 24          0024  1 !*                                                                        *
; 25          0025  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS *
; 26          0026  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                *
; 27          0027  1 !*                                                                        *
; 28          0028  1 !*                                                                        *
; 29          0029  1 !**************************************************************************
```

```
;    31          0030  1 !++
;    32          0031  1 !
;    33          0032  1 ! Facility:     VAX-11 CONVERT
;    34          0033  1 !
;    35          0034  1 ! Abstract:     RMS file handeling routines
;    36          0035  1 !
;    37          0036  1 ! Contents:
;    38          0037  1 !                PARSE_DEF
;    39          0038  1 !                OPEN_INPUT
;    40          0039  1 !                SEARCH_FILE
;    41          0040  1 !                OPEN_IN
;    42          0041  1 !                OPEN_OUTPUT
;    43          0042  1 !                GET_PROLOGUE
;    44          0043  1 !                CREATE_BUFFER
;    45          0044  1 !
;    46          0045  1 ! Environment:
;    47          0046  1 !
;    48          0047  1 !                VAX/VMS Operating System
;    49          0048  1 !
;    50          0049  1 !
;    51          0050  1 ! Author:       Keith B. Thompson       Creation date:  June-1980
;    52          0051  1 !
;    53          0052  1 !
;    54          0053  1 ! Modified by:
;    55          0054  1 !
;    56          0055  1 !       V03-013 JWT0194         Jim Teague              31-Aug-1984
;    57          0056  1 !               Fix problem with CONVERT dropping blocks when input
;    58          0057  1 !               file is UDF.
;    59          0058  1 !
;    60          0059  1 !       V03-012 RAS0311         Ron Schaefer            18-Jun-1984
;    61          0060  1 !               Fix output file related file parsing by making sure the
;    62          0061  1 !               input file result filespec is available.  Fix to RAS0260.
;    63          0062  1 !
;    64          0063  1 !       V03-011 RAS0272         Ron Schaefer            16-Mar-1984
;    65          0064  1 !               Allow CONVERT to fastload & sort network files since
;    66          0065  1 !               SORT-32 is now abel to handle them.
;    67          0066  1 !
;    68          0067  1 !       V03-010 RAS0260         Ron Schaefer            2-Mar-1984
;    69          0068  1 !               Improve performance of RAS0250 by copying the DVI, FID and
;    70          0069  1 !               DID fields from the LIB$FIND_FILE NAM to the real NAM
;    71          0070  1 !               used for the open.  Also copy the device characteristics.
;    72          0071  1 !
;    73          0072  1 !       V03-009 RAS0250         Ron Schaefer            23-Feb-1984
;    74          0073  1 !               Convert SEARCH_FILE to use LIB$FIND_FILE for correct
;    75          0074  1 !               related file processing.   Add FDL_STRING support.
;    76          0075  1 !
;    77          0076  1 !       V03-008 KBT0442         Keith B. Thompson       30-Dec-1982
;    78          0077  1 !               Make fdl_fab/rab global
;    79          0078  1 !
;    80          0079  1 !       V03-007 KBT0435         Keith B. Thompson       16-Dec-1982
;    81          0080  1 !               Always open the input file to fill the fab
;    82          0081  1 !               except when comming from tape and sorting
;    83          0082  1 !
;    84          0083  1 !       V03-006 KBT0392         Keith B. Thompson       29-Oct-1982
;    85          0084  1 !               Call new read_prologue routine
;    86          0085  1 !
;    87          0086  1 !       V03-005 KBT0370         Keith B. Thompson       19-Oct-1982
```

```
;    88          0087  1 !                 Use new supported fdl$parse
;    89          0088  1 !
;    90          0089  1 !       V03-004 KBT0347         Keith B. Thompson       4-Oct-1982
;    91          0090  1 !                 Use new linkage definitions
;    92          0091  1 !
;    93          0092  1 !       V03-003 KBT0044         Keith Thompson          5-Apr-1982
;    94          0093  1 !                 Don't do a search on a device mounted foreign
;    95          0094  1 !
;    96          0095  1 !       V03-002 KBT0025         Keith Thompson          26-Mar-1982
;    97          0096  1 !                 Fix fill switch for /nofast
;    98          0097  1 !
;    99          0098  1 !       V03-001 KBT0015         Keith Thompson          18-Mar-1982
;   100          0099  1 !                 Fix area allocation bug in get_prologue and use new
;   101          0100  1 !                 plg$c_ver3 instead of literal
;   102          0101  1 !
;   103          0102  1 !****
```

```
;  105          0103  1
;  106          0104  1 PSECT
;  107          0105  1         OWN     = _CONV$OWN     (PIC),
;  108          0106  1         GLOBAL  = _CONV$GLOBAL  (PIC),
;  109          0107  1         PLIT    = _CONV$PLIT    (SHARE,PIC),
;  110          0108  1         CODE    = _CONV$CODE    (SHARE,PIC);
;  111          0109  1
;  112          0110  1 LIBRARY 'SYS$LIBRARY:LIB.L32';
;  113          0111  1 LIBRARY 'SRC$:CONVERT';
;  114          0112  1
;  115          0113  1 EXTERNAL ROUTINE
;  116          0114  1         CONV$$GET_VM            : CL$GET_VM,
;  117          0115  1         CONV$$READ_PROLOGUE     : CL$READ_PROLOGUE      NOVALUE,
;  118          0116  1         CONV$$RMS_OPEN_ERROR    : NOVALUE,
;  119          0117  1         FDL$PARSE               : ADDRESSING_MODE( GENERAL ),
;  120          0118  1         LIB$FIND_FILE           : ADDRESSING_MODE( GENERAL );
;  121          0119  1
;  122          0120  1 FORWARD ROUTINE
;  123          0121  1         CONV$$SEARCH_FILE,
;  124          0122  1         CONV$$OPEN_IN;
;  125          0123  1
;  126          0124  1 !       Error codes
;  127          0125  1 !
;  128          0126  1 DEFINE_ERROR_CODES;
;  129          0127  1
;  130          0128  1 EXTERNAL
;  131          0129  1
;  132          0130  1 !       The Option Flags:
;  133          0131  1 !
;  134          0132  1         CONV$GL_APPEND          : LONG,                 ! APPEND
;  135          0133  1         CONV$GL_CREATE          : LONG,                 ! CREATE
;  136          0134  1         CONV$GL_FDL             : LONG,                 ! FDL
;  137          0135  1         CONV$GL_EXC             : LONG,                 ! EXCEPTION
;  138          0136  1         CONV$GL_FAST            : LONG,                 ! FAST
;  139          0137  1         CONV$GL_MERGE           : LONG,                 ! MERGE
;  140          0138  1         CONV$GL_FILL            : LONG,                 ! FILL_BUCKETS
;  141          0139  1         CONV$GL_FIX             : LONG,                 ! FIXED_WRITE
;  142          0140  1         CONV$GL_KEY             : LONG,                 ! KEY
;  143          0141  1         CONV$GL_PAD             : LONG,                 ! PAD_RECORDS
;  144          0142  1         CONV$GL_SHARE           : LONG,                 ! SHARE
;  145          0143  1         CONV$GL_SORT            : LONG,                 ! SORT
;  146          0144  1         CONV$GL_READ_C          : LONG,                 ! READ_CHECK
;  147          0145  1         CONV$GL_TRUNCATE        : LONG,                 ! TRUNCATE
;  148          0146  1         CONV$GL_WRITE_C         : LONG,                 ! WRITE_CHECK
;  149          0147  1         CONV$GL_PROLOG          : LONG,                 ! PROLOGUE
;  150          0148  1         CONV$AB_FLAGS           : BLOCK [ ,BYTE ],
;  151          0149  1
;  152          0150  1         CONV$GW_OUT_MRS         : WORD,
;  153          0151  1         CONV$GW_UDF_MRS         : WORD,
;  154          0152  1         CONV$GB_CURRENT_FILE    : BYTE,
;  155          0153  1         CONV$GW_MAX_REC_SIZ     : WORD,
;  156          0154  1         CONV$GL_REC_BUF_PTR,
;  157          0155  1         CONV$GL_VFC_BUF_PTR,
;  158          0156  1         CONV$GL_FINDFILE_CTX,
;  159          0157  1
;  160          0158  1         CONV$AL_IN_FILE_NAM     : VECTOR [ ,LONG ],             ! Input File
;  161          0159  1         CONV$AR_OUT_FILE_NAM    : REF DESC_BLK,                 ! Output File
```

```
;  162          0160  1          CONV$AR_FDL_FILE_NAM     : REF DESC_BLK,                 ! FDL File
;  163          0161  1
;  164          0162  1          CONV$AB_IN_XABSUM        : $XABSUM_DECL,
;  165          0163  1          CONV$AB_IN_XABFHC        : $XABFHC_DECL,
;  166          0164  1          CONV$AB_IN_NAM           : $NAM_DECL,
;  167          0165  1          CONV$AB_IN_FAB           : $FAB_DECL,
;  168          0166  1          CONV$AB_IN_RAB           : $RAB_DECL,
;  169          0167  1          CONV$AB_OUT_XABSUM       : $XABSUM_DECL,
;  170          0168  1          CONV$AB_OUT_NAM          : $NAM_DECL,
;  171          0169  1          CONV$AB_OUT_FAB          : $FAB_DECL,
;  172          0170  1          CONV$AB_OUT_RAB          : $RAB_DECL;
;  173          0171  1
;  174          0172  1 GLOBAL
;  175          0173  1          CONV$AB_FDL_FAB          : REF BLOCK [ ,BYTE ],
;  176          0174  1          CONV$AB_FDL_RAB          : REF BLOCK [ ,BYTE ];
;  177          0175  1
```

```
;  179          0176  1 %SBTTL  'PARSE_DEF'
;  180          0177  1 GLOBAL ROUTINE CONV$$PARSE_DEF =
;  181          0178  1 !++
;  182          0179  1 !
;  183          0180  1 ! Functional Description:
;  184          0181  1 !
;  185          0182  1 !       Calls fdl$parse to parse the fdl file and fill in a fab.  The
;  186          0183  1 !       info from this fab is will be copied to the output fab in open_output
;  187          0184  1 !
;  188          0185  1 ! Calling Sequence:
;  189          0186  1 !
;  190          0187  1 !       CONV$$PARSE()
;  191          0188  1 !
;  192          0189  1 ! Input Parameters:
;  193          0190  1 !       none
;  194          0191  1 !
;  195          0192  1 ! Implicit Inputs:
;  196          0193  1 !
;  197          0194  1 !       CONV$AR_FDL_FILE_NAME   - FDL file descriptor
;  198          0195  1 !
;  199          0196  1 ! Output Parameters:
;  200          0197  1 !       none
;  201          0198  1 !
;  202          0199  1 ! Implicit Outputs:
;  203          0200  1 !       none
;  204          0201  1 !
;  205          0202  1 ! Routine Value:
;  206          0203  1 !
;  207          0204  1 !       Value returned by fdl$parse
;  208          0205  1 !
;  209          0206  1 ! Routines Called:
;  210          0207  1 !
;  211          0208  1 !       FDL$PARSE
;  212          0209  1 !
;  213          0210  1 ! Side Effects:
;  214          0211  1 !
;  215          0212  1 !--
;  216          0213  1
;  217          0214  2     BEGIN
;  218          0215  2
;  219          0216  2 !    EXTERNAL LITERAL
;  220          0217  2 !       FDL$M_FDL_STRING,
;  221          0218  2 !       FDL$M_SIGNAL;
;  222          0219  2
;  223          0220  2     LOCAL
;  224          0221  2         FDL_FLAGS       : LONG;
;  225          0222  2
;  226          0223  2     ! Initialize the flags
;  227          0224  2     !
;  228          0225  2     FDL_FLAGS = 0;
;  229          0226  2
;  230          0227  2     ! If convert is signaling then fdl should
;  231          0228  2     !
;  232          0229  2     IF .CONV$AB_FLAGS [ CONV$V_SIGNAL ]
;  233          0230  2     THEN
;  234          0231  2 !       FDL_FLAGS = FDL$M_SIGNAL;
;  235          0232  2         FDL_FLAGS = 1;
```

```
;  236     0233  2
;  237     0234  2      ! If caller passed in an fdl string, then tell fdl about it
;  238     0235  2      !
;  239     0236  2      IF .CONV$AB_FLAGS [ CONV$V_FDL_STRING ]
;  240     0237  2      THEN
;  241     0238  2  !       FDL_FLAGS = FDL$M_FDL_STRING OR .FDL_FLAGS;
;  242     0239  2          FDL_FLAGS = 2 OR .FDL_FLAGS;
;  243     0240  2
;  244     0241  2
;  245     0242  2      RETURN FDL$PARSE( .CONV$AR_FDL_FILE_NAM,
;  246     0243  2                        CONV$AB_FDL_FAB,
;  247     0244  2                        CONV$AB_FDL_RAB,
;  248     0245  2                        FDL_FLAGS )
;  249     0246  2
;  250     0247  1      END;
```

```
                                        .TITLE  CONV$FILES VAX-11 CONVERT
                                        .IDENT  \V04-000\

                                        .PSECT  _CONV$GLOBAL,NOEXE,  PIC,2

                                00000 CONV$AB_FDL_FAB::
                                        .BLKB   4
                                00004 CONV$AB_FDL_RAB::
                                        .BLKB   4

                                        .EXTRN  CONV$$GET_VM, CONV$$READ_PROLOGUE
                                        .EXTRN  CONV$$RMS_OPEN_ERROR
                                        .EXTRN  FDL$PARSE, LIB$FIND_FILE
                                        .EXTRN  CONVERT$_FACILITY
                                        .EXTRN  CONV$_FAO_MAX, CONV$_BADBLK
                                        .EXTRN  CONV$_BADLOGIC, CONV$_BADSORT
                                        .EXTRN  CONV$_CONFQUAL, CONV$_CREATEDSTM
                                        .EXTRN  CONV$_CREA_ERR, CONV$_DELPRI
                                        .EXTRN  CONV$_DUP, CONV$_EXTN_ERR
                                        .EXTRN  CONV$_FATALEXC, CONV$_FILLIM
                                        .EXTRN  CONV$_IDX_LIM, CONV$_ILL_KEY
                                        .EXTRN  CONV$_ILL_VALUE
                                        .EXTRN  CONV$_INP_FILES
                                        .EXTRN  CONV$_INSVIRMEM
                                        .EXTRN  CONV$_INVBKT, CONV$_KEY
                                        .EXTRN  CONV$_KEYREF, CONV$_LOADIDX
                                        .EXTRN  CONV$_NARG, CONV$_NI
                                        .EXTRN  CONV$_NOKEY, CONV$_NOTIDX
                                        .EXTRN  CONV$_NOTSEQ, CONV$_NOWILD
                                        .EXTRN  CONV$_ORDER, CONV$_OPENEXC
                                        .EXTRN  CONV$_OPENIN, CONV$_OPENOUT
                                        .EXTRN  CONV$_PAD, CONV$_PLV
                                        .EXTRN  CONV$_PROERR, CONV$_PROL_WRT
                                        .EXTRN  CONV$_READERR, CONV$_RSK
                                        .EXTRN  CONV$_RSZ, CONV$_RTL
                                        .EXTRN  CONV$_RTS, CONV$_SEQ
                                        .EXTRN  CONV$_UDF_BKS, CONV$_UDF_BLK
                                        .EXTRN  CONV$_VFC, CONV$_WRITEERR
                                        .EXTRN  CONV$GL_APPEND, CONV$GL_CREATE
                                        .EXTRN  CONV$GL_FDL, CONV$GL_EXC
```

```
                                        .EXTRN  CONV$GL_FAST, CONV$GL_MERGE
                                        .EXTRN  CONV$GL_FILL, CONV$GL_FIX
                                        .EXTRN  CONV$GL_KEY, CONV$GL_PAD
                                        .EXTRN  CONV$GL_SHARE, CONV$GL_SORT
                                        .EXTRN  CONV$GL_READ_C, CONV$GL_TRUNCATE
                                        .EXTRN  CONV$GL_WRITE_C
                                        .EXTRN  CONV$GL_PROLOG, CONV$AB_FLAGS
                                        .EXTRN  CONV$GW_OUT_MRS
                                        .EXTRN  CONV$GW_UDF_MRS
                                        .EXTRN  CONV$GB_CURRENT_FILE
                                        .EXTRN  CONV$GW_MAX_REC_SIZ
                                        .EXTRN  CONV$GL_REC_BUF_PTR
                                        .EXTRN  CONV$GL_VFC_BUF_PTR
                                        .EXTRN  CONV$GL_FINDFILE_CTX
                                        .EXTRN  CONV$AL_IN_FILE_NAM
                                        .EXTRN  CONV$AR_OUT_FILE_NAM
                                        .EXTRN  CONV$AR_FDL_FILE_NAM
                                        .EXTRN  CONV$AB_IN_XABSUM
                                        .EXTRN  CONV$AB_IN_XABFHC
                                        .EXTRN  CONV$AB_IN_NAM, CONV$AB_IN_FAB
                                        .EXTRN  CONV$AB_IN_RAB, CONV$AB_OUT_XABSUM
                                        .EXTRN  CONV$AB_OUT_NAM
                                        .EXTRN  CONV$AB_OUT_FAB
                                        .EXTRN  CONV$AB_OUT_RAB

                                        .PSECT  _CONV$CODE,NOWRT,  SHR,  PIC,2

                          0000 00000    .ENTRY  CONV$$PARSE_DEF, Save nothing                   ; 0177
                       7E   D4 00002    CLRL    FDL_FLAGS                                       ; 0225
                 03    0000G  CF  E9 00004    BLBC    CONV$AB_FLAGS, 1$                       ; 0229
                 6E           01  D0 00009    MOVL    #1, FDL_FLAGS                           ; 0232
           03    0000G  CF           01  E1 0000C 1$:  BBC     #1, CONV$AB_FLAGS, 2$           ; 0236
                 6E           02  88 00012    BISB2   #2, FDL_FLAGS                           ; 0239
                              5E  DD 00015 2$:  PUSHL   SP                                      ; 0242
                0000'  CF  9F 00017    PUSHAB  CONV$AB_FDL_RAB                                 ;
                0000'  CF  9F 0001B    PUSHAB  CONV$AB_FDL_FAB                                 ;
                0000G  CF  DD 0001F    PUSHL   CONV$AR_FDL_FILE_NAM                            ;
          00000000G  00        04  FB 00023    CALLS   #4, FDL$PARSE                           ;
                               04 0002A    RET                                             ; 0247

; Routine Size:  43 bytes,    Routine Base:  _CONV$CODE + 0000
```

```
;   252     0248  1 %SBTTL  'OPEN_INPUT'
;   253     0249  1 GLOBAL ROUTINE  CONV$$OPEN_INPUT =
;   254     0250  1 !++
;   255     0251  1 !
;   256     0252  1 ! Functional Description:
;   257     0253  1 !
;   258     0254  1 !       Opens an input file
;   259     0255  1 !
;   260     0256  1 ! Calling Sequence:
;   261     0257  1 !
;   262     0258  1 !       CONV$$OPEN_INPUT()
;   263     0259  1 !
;   264     0260  1 ! Input Parameters:
;   265     0261  1 !       none
;   266     0262  1 !
;   267     0263  1 ! Implicit Inputs:
;   268     0264  1 !       none
;   269     0265  1 !
;   270     0266  1 ! Output Parameters:
;   271     0267  1 !       none
;   272     0268  1 !
;   273     0269  1 ! Implicit Outputs:
;   274     0270  1 !       none
;   275     0271  1 !
;   276     0272  1 ! Routine Value:
;   277     0273  1 !
;   278     0274  1 !       CONV$_SUCCESS or error code from CONV$$SEARCH_FILE or CONV$$OPEN_IN
;   279     0275  1 !
;   280     0276  1 ! Routines Called:
;   281     0277  1 !
;   282     0278  1 !       CONV$$SEARCH_FILE
;   283     0279  1 !       CONV$$OPEN_IN
;   284     0280  1 !
;   285     0281  1 ! Side Effects:
;   286     0282  1 !
;   287     0283  1 !       Opens an input file
;   288     0284  1 !
;   289     0285  1 !--
;   290     0286  1
;   291     0287  2     BEGIN
;   292     0288  2
;   293     0289  2     LOCAL
;   294     0290  2         STATUS  : LONG,
;   295     0291  2         IN_DEV  : BLOCK [ 1,LONG ];
;   296     0292  2
;   297     0293  2
;   298     0294  2     ! Any errors on the input fab are OPENIN errors
;   299     0295  2     !
;   300     0296  2     CONV$AB_IN_FAB [ FAB$L_CTX ] = CONV$_OPENIN;
;   301     0297  2
;   302     0298  2     ! Start by getting the file name
;   303     0299  2     !
;   304     0300  2     RET_ON_ERROR( CONV$$SEARCH_FILE() );
;   305     0301  2
;   306     0302  2     ! For now there are only sequential files on tape if there is no
;   307     0303  2     ! definition file then it cant be a fast load
;   308     0304  2     !
```

```
;  309          0305  2     IN_DEV = .CONV$AB_IN_FAB [ FAB$L_DEV ];
;  310          0306  2
;  311          0307  2     IF ( NOT .CONV$GL_FDL ) AND .IN_DEV [ DEV$V_SQD ]
;  312          0308  2     THEN
;  313          0309  2         BEGIN
;  314          0310  2         CONV$GL_FAST    = _CLEAR;
;  315          0311  2         CONV$GL_SORT    = _CLEAR
;  316          0312  2         END;
;  317          0313  2
;  318          0314  2     ! If we are sorting the input file from tape or network
;  319          0315  2     ! then don't bother opening the input file here
;  320          0316  2     !
;  321          0317  2     IF .CONV$GL_SORT AND (.IN_DEV [ DEV$V_SQD ] OR .IN_DEV [ DEV$V_NET ])
;  322          0318  2     THEN
;  323          0319  2         RETURN CONV$_SUCCESS
;  324          0320  2     ELSE
;  325          0321  2         RETURN CONV$$OPEN_IN()
;  326          0322  2
;  327          0323  1     END;



                                    0000 00000          .ENTRY  CONV$$OPEN_INPUT, Save nothing                  ; 0249
                0000G  CF 00000000G 8F   D0 00002       MOVL    #CONV$_OPENIN, CONV$AB_IN_FAB+24                ; 0296
                0000V  CF           00   FB 0000B       CALLS   #0, CONV$$SEARCH_FILE                           ; 0300
                       2C           50   E9 00010       BLBC    STATUS, 4$                                      ;
                       50     0000G CF   D0 00013       MOVL    CONV$AB_IN_FAB+64, IN_DEV                       ; 0305
                       0C     0000G CF   E8 00018       BLBS    CONV$GL_FDL, 1$                                 ; 0307
                08     50           05   E1 0001D       BBC     #5, IN_DEV, 1$                                  ;
                              0000G CF   D4 00021       CLRL    CONV$GL_FAST                                    ; 0310
                              0000G CF   D4 00025       CLRL    CONV$GL_SORT                                    ; 0311
                       0C     0000G CF   E9 00029 1$:   BLBC    CONV$GL_SORT, 3$                                ; 0317
                04     50           05   E0 0002E       BBS     #5, IN_DEV, 2$                                  ;
                04     50           0D   E1 00032       BBC     #13, IN_DEV, 3$                                 ;
                       50           01   D0 00036 2$:   MOVL    #1, R0                                          ; 0321
                                         04 00039       RET                                                     ;
                0000V  CF           00   FB 0003A 3$:   CALLS   #0, CONV$$OPEN_IN                               ;
                                         04 0003F 4$:   RET                                                     ; 0323

; Routine Size:  64 bytes,    Routine Base:  _CONV$CODE + 002B
```

```
;  329          0324  1 %SBTTL  'SEARCH_FILE'
;  330          0325  1 GLOBAL ROUTINE CONV$$SEARCH_FILE =
;  331          0326  1 !++
;  332          0327  1 !
;  333          0328  1 ! Functional Description:
;  334          0329  1 !
;  335          0330  1 !       Searches for an input file
;  336          0331  1 !
;  337          0332  1 ! Calling Sequence:
;  338          0333  1 !
;  339          0334  1 !       CONV$$SEARCH_FILE()
;  340          0335  1 !
;  341          0336  1 ! Input Parameters:
;  342          0337  1 !       none
;  343          0338  1 !
;  344          0339  1 ! Implicit Inputs:
;  345          0340  1 !
;  346          0341  1 !       CONV$AL_IN_FILE_NAME    - Input file name descriptor array
;  347          0342  1 !       CONV$GB_CURRENT_FILE    - Current input file name being searched
;  348          0343  1 !       CONV$GL_FINDFILE_CTX    - LIB$FIND_FILE context
;  349          0344  1 !
;  350          0345  1 ! Output Parameters:
;  351          0346  1 !       none
;  352          0347  1 !
;  353          0348  1 ! Implicit Outputs:
;  354          0349  1 !       none
;  355          0350  1 !
;  356          0351  1 ! Routine Value:
;  357          0352  1 !
;  358          0353  1 !       CONV$_SUCCESS or CONV$_NOWILD
;  359          0354  1 !
;  360          0355  1 ! Routines Called:
;  361          0356  1 !
;  362          0357  1 !       LIB$FIND_FILE
;  363          0358  1 !       $PARSE
;  364          0359  1 !       CONV$$RMS_OPEN_ERROR    - By RMS as an AST or by us on LIB$FIND_FILE errors
;  365          0360  1 !       $SEARCH
;  366          0361  1 !
;  367          0362  1 ! Side Effects:
;  368          0363  1 !
;  369          0364  1 !       Sets up input name block for next input file
;  370          0365  1 !
;  371          0366  1 !--
;  372          0367  1
;  373          0368  2     BEGIN
;  374          0369  2
;  375          0370  2     LOCAL
;  376          0371  2         STATUS,
;  377          0372  2         STV,
;  378          0373  2         FINDFILENAM     : REF BLOCK [ ,BYTE ],
;  379          0374  2         IN_NAME         : REF DESC_BLK,
;  380          0375  2         OUT_NAME        : DESC_BLK,
;  381          0376  2         IN_DEVICE       : BLOCK [ 1,LONG ];
;  382          0377  2
;  383          0378  2     BIND
;  384          0379  2         FINDFILEFAB     = CONV$GL_FINDFILE_CTX : REF BLOCK[ ,BYTE];
;  385          0380  2
```

```
;  386          0381  2      IN_NAME = .CONV$AL_IN_FILE_NAM [ .CONV$GB_CURRENT_FILE ];
;  387          0382  2
;  388          0383  2      OUT_NAME [DSC$B_CLASS] = DSC$K_CLASS_D;
;  389          0384  2      OUT_NAME [DSC$B_DTYPE] = DSC$K_DTYPE_T;
;  390          0385  2      OUT_NAME [DSC$W_LENGTH] = 0;
;  391          0386  2      OUT_NAME [DSC$A_POINTER] = 0;
;  392          0387  2
;  393          0388  2      ! Get the next file name to search for
;  394          0389  2      !
;  395          0390  2      STATUS = LIB$FIND_FILE(
;  396          0391  2                  .IN_NAME, OUT_NAME,
;  397          0392  2                  CONV$GL_FINDFILE_CTX,
;  398          0393  2                  0, 0, STV, %REF(3));
;  399          0394  2
;  400          0395  2      ! If the filename has wildcards in it it's an error
;  401          0396  2      !
;  402          0397  2      IF (.STATUS AND STS$M_MSG_NO)  EQL  SHR$_NOWILD
;  403          0398  2      THEN
;  404          0399  2          RETURN CONV$_NOWILD;
;  405          0400  2
;  406          0401  2      ! Report miscellaneous errors from LIB$FIND_FILE
;  407          0402  2      !
;  408          0403  2      IF NOT .STATUS
;  409          0404  2      THEN
;  410          0405  3          BEGIN
;  411          0406  3          FINDFILEFAB [ FAB$L_CTX ] = CONV$_OPENIN;
;  412          0407  3          CONV$$RMS_OPEN_ERROR(.FINDFILEFAB);
;  413          0408  2          END;
;  414          0409  2
;  415          0410  2      CONV$AB_IN_FAB [ FAB$B_FNS ] = .OUT_NAME [ DSC$W_LENGTH ];
;  416          0411  2      CONV$AB_IN_FAB [ FAB$L_FNA ] = .OUT_NAME [ DSC$A_POINTER ];
;  417          0412  2
;  418          0413  2      ! Clear the IFI and device char. so we can parse
;  419          0414  2      !
;  420          0415  2      CONV$AB_IN_FAB [ FAB$W_IFI ] = 0;
;  421          0416  2      CONV$AB_IN_FAB [ FAB$L_DEV ] = .FINDFILEFAB [ FAB$L_DEV ];
;  422          0417  2
;  423          0418  2      FINDFILENAM = .FINDFILEFAB [ FAB$L_NAM ];
;  424          0419  2
;  425          0420  2      ! Copy the DVI, FID and DID fields to the NAM block to use for the open.
;  426          0421  2      !
;  427          0422  2      CH$MOVE( NAM$S_DVI+NAM$S_FID+NAM$S_DID,
;  428          0423  2          FINDFILENAM [ NAM$T_DVI ], CONV$AB_IN_NAM [ NAM$T_DVI ]);
;  429          0424  2
;  430          0425  2      RETURN CONV$_SUCCESS
;  431          0426  2
;  432          0427  1      END;


                                 007C 00000          .ENTRY  CONV$$SEARCH_FILE, Save R2,R3,R4,R5,R6      ; 0325
                    56     0000G  CF   9E 00002      MOVAB   CONV$GL_FINDFILE_CTX, R6                    ;
                    5E            0C   C2 00007      SUBL2   #12, SP                                     ;
                    50     0000G  CF   9A 0000A      MOVZBL  CONV$GB_CURRENT_FILE, R0                    ; 0381
                    50     0000GCF40   D0 0000F      MOVL    CONV$AL_IN_FILE_NAM[R0], IN_NAME            ;
```

```
                          04    AE 020E0000      8F   D0 00015          MOVL    #34471936, OUT_NAME                     ;0385
                                           08  AE   D4 0001D          CLRL    OUT_NAME+4                              ;0386
                                               03   DD 00020          PUSHL   #3                                      ;0393
                                               5E   DD 00022          PUSHL   SP
                                           08  AE   9F 00024          PUSHAB  STV                                     ;0390
                                               7E   7C 00027          CLRQ    -(SP)
                                               56   DD 00029          PUSHL   R6
                                           1C  AE   9F 0002B          PUSHAB  OUT_NAME
                                               50   DD 0002E          PUSHL   IN_NAME                                 ;0391
                              00000000G    00        07   FB 00030    CALLS   #7, LIB$FIND_FILE
                                           51        50   D0 00037    MOVL    R0, STATUS
                          50              51 FFFF0007  8F   CB 0003A    BICL3   #-65529, STATUS, R0                     ;0397
                              00001128    8F        50   D1 00042    CMPL    R0, #4392
                                                    08   12 00049    BNEQ    1$
                                           50 00000000G 8F   D0 0004B    MOVL    #CONV$_NOWILD, R0                       ;0399
                                                    04   00052    RET
                                           12        51   E8 00053 1$: BLBS    STATUS, 2$                              ;0403
                                           50        66   D0 00056    MOVL    FINDFILEFAB, R0                         ;0406
                                18         A0 00000000G 8F   D0 00059    MOVL    #CONV$_OPENIN, 24(R0)
                                                    50   DD 00061    PUSHL   R0                                      ;0407
                             0000G         CF        01   FB 00063    CALLS   #1, CONV$$RMS_OPEN_ERROR
                             0000G         CF     08 AE   90 00068 2$: MOVB    OUT_NAME, CONV$AB_IN_FAB+52             ;0410
                             0000G         CF     0C AE   D0 0006E    MOVL    OUT_NAME+4, CONV$AB_IN_FAB+44           ;0411
                                         0000G    CF   B4 00074    CLRW    CONV$AB_IN_FAB+2                        ;0415
                                           50        66   D0 00078    MOVL    FINDFILEFAB, R0                         ;0416
                             0000G         CF     40 A0   D0 0007B    MOVL    64(R0), CONV$AB_IN_FAB+64
                                           50     28 A0   D0 00081    MOVL    40(R0), FINDFILENAM                     ;0418
                  0000G  CF        14      A0        1C   28 00085    MOVC3   #28, 20(FINDFILENAM), CONV$AB_IN_NAM+20  ;0423
                                           50        01   D0 0008C    MOVL    #1, R0                                  ;0425
                                                    04   0008F    RET                                             ;0427

; Routine Size:  144 bytes,    Routine Base:  _CONV$CODE + 006B

;  433          0428  1
```

```
;  435          0429  1 %SBTTL  'OPEN_IN'
;  436          0430  1 GLOBAL ROUTINE  CONV$$OPEN_IN =
;  437          0431  1 !++
;  438          0432  1 !
;  439          0433  1 ! Functional Description:
;  440          0434  1 !
;  441          0435  1 !       Actually does the open of the input file, allocates and fills
;  442          0436  1 !       in key and area xabs if necessary, also connects record stream
;  443          0437  1 !
;  444          0438  1 ! Calling Sequence:
;  445          0439  1 !
;  446          0440  1 !       CONV$$OPEN_IN()
;  447          0441  1 !
;  448          0442  1 ! Input Parameters:
;  449          0443  1 !       none
;  450          0444  1 !
;  451          0445  1 ! Implicit Inputs:
;  452          0446  1 !
;  453          0447  1 !       CONV$AB_IN_FAB  - Input fab
;  454          0448  1 !
;  455          0449  1 ! Output Parameters:
;  456          0450  1 !       none
;  457          0451  1 !
;  458          0452  1 ! Implicit Outputs:
;  459          0453  1 !       none
;  460          0454  1 !
;  461          0455  1 ! Routine Value:
;  462          0456  1 !
;  463          0457  1 !       CONV$_SUCCESS or CONV$_NOKEY
;  464          0458  1 !
;  465          0459  1 ! Routines Called:
;  466          0460  1 !
;  467          0461  1 !       $OPEN
;  468          0462  1 !       CONV$$RMS_OPEN_ERROR    - By RMS as an AST
;  469          0463  1 !       CONV$$GET_VM
;  470          0464  1 !       $DISPLAY
;  471          0465  1 !       $CONNECT
;  472          0466  1 !
;  473          0467  1 ! Side Effects:
;  474          0468  1 !
;  475          0469  1 !       Opens and connects input file
;  476          0470  1 !
;  477          0471  1 !--
;  478          0472  1
;  479          0473  2     BEGIN
;  480          0474  2
;  481          0475  2     LOCAL
;  482          0476  2         STATUS  : LONG;
;  483          0477  2
;  484          0478  2     ! Set the FAB from the Option Switches
;  485          0479  2     !
;  486          0480  2     ! Read Check
;  487          0481  2     !
;  488          0482  2     CONV$AB_IN_FAB [ FAB$V_RCK ] = .CONV$GL_READ_C;
;  489          0483  2
;  490          0484  2     ! Input file sharing
;  491          0485  2     !
```

```
;  492          0486  2      IF .CONV$GL_SHARE
;  493          0487  2      THEN
;  494          0488  2          BEGIN
;  495          0489  2
;  496          0490  2          ! Set up the file sharing bits
;  497          0491  2          !
;  498          0492  2          CONV$AB_IN_FAB [ FAB$B_SHR ] = FAB$M_PUT OR FAB$M_GET OR FAB$M_DEL OR
;  499          0493  2                                                         FAB$M_UPD OR FAB$M_UPI;
;  500          0494  2
;  501          0495  2          ! Do not wait for any record locks
;  502          0496  2          !
;  503          0497  2          CONV$AB_IN_RAB [ RAB$V_RRL ] = _SET
;  504          0498  2
;  505          0499  2          END;
;  506          0500  2
;  507          0501  2      ! If we have to access the file by key (other then primary) or we have
;  508          0502  2      ! to sort the file (which means we use RFA access)
;  509          0503  2      ! then clear the sqo bit
;  510          0504  2      !
;  511          0505  2      IF ( .CONV$GL_KEY NEQU 0 ) OR .CONV$GL_SORT
;  512          0506  2      THEN
;  513          0507  2          CONV$AB_IN_FAB [ FAB$V_SQO ] = _CLEAR;
;  514          0508  2
;  515          0509  2      ! Open the file
;  516          0510  2      !
;  517          0511  2      $OPEN ( FAB=CONV$AB_IN_FAB,ERR=CONV$$RMS_OPEN_ERROR );
;  518          0512  2
;  519          0513  2      ! Say that the file is open
;  520          0514  2      !
;  521          0515  2      CONV$AB_FLAGS [ CONV$V_IN ] = _SET;
;  522          0516  2
;  523          0517  2      ! If this is an index file and we are creating the output file not by
;  524          0518  2      ! FDL definition then get the area ad key xabs
;  525          0519  2      !
;  526          0520  2      IF ( .CONV$AB_IN_FAB [ FAB$B_ORG ] EQLU FAB$C_IDX ) AND
;  527          0521  2                                  .CONV$GL_CREATE AND ( NOT .CONV$GL_FDL )
;  528          0522  2      THEN
;  529          0523  2          BEGIN
;  530          0524  2
;  531          0525  2          LOCAL
;  532          0526  2                  BYTES,
;  533          0527  2                  VM_POINTER,
;  534          0528  2                  CURRENTXAB      : REF BLOCK [ ,BYTE ];
;  535          0529  2
;  536          0530  2          BIND
;  537          0531  2                  NEWXAB  = VM_POINTER : REF BLOCK [ ,BYTE ];
;  538          0532  2
;  539          0533  2          ! Find out how much memory we need (The extra 32 is for the key name buffer)
;  540          0534  2          !
;  541          0535  2          BYTES = .CONV$AB_IN_XABSUM [ XAB$B_NOK ] * ( XAB$C_KEYLEN + 32 );
;  542          0536  2          BYTES = ( .CONV$AB_IN_XABSUM [ XAB$B_NOA ] * XAB$C_ALLLEN ) + .BYTES;
;  543          0537  2
;  544          0538  2          ! Get the address space
;  545          0539  2          !
;  546          0540  2          VM_POINTER = CONV$$GET_VM ( .BYTES  );
;  547          0541  2
;  548          0542  3          ! The protection xab will point to the new xabs
```

```
;  549          0543  1          !
;  550          0544  1          CURRENTXAB = CONV$AB_IN_XABSUM;
;  551          0545  1          !
;  552          0546  1          ! Chain the xabs together and set up the fields
;  553          0547  1          ! Keys first
;  554          0548  1          !
;  555          0549  1          INCR I FROM 0 TO .CONV$AB_IN_XABSUM [ XAB$B_NOK ] - 1 BY 1
;  556          0550  1          DO
;  557          0551  1              BEGIN
;  558          0552  1              CURRENTXAB [ XAB$L_NXT ] = .NEWXAB;
;  559          0553  1              CURRENTXAB = .NEWXAB;
;  560          0554  1              CURRENTXAB [ XAB$B_COD ] = XAB$C_KEY;
;  561          0555  1              CURRENTXAB [ XAB$B_BLN ] = XAB$C_KEYLEN;
;  562          0556  1              CURRENTXAB [ XAB$B_REF ] = .I;
;  563          0557  1              CURRENTXAB [ XAB$L_KNM ] = .CURRENTXAB + XAB$C_KEYLEN;
;  564          0558  1              NEWXAB = .NEWXAB + XAB$C_KEYLEN + 32
;  565          0559  1              END;
;  566          0560  1          !
;  567          0561  1          ! Then areas
;  568          0562  1          !
;  569          0563  1          INCR I FROM 0 TO .CONV$AB_IN_XABSUM [ XAB$B_NOA ] - 1 BY 1
;  570          0564  1          DO
;  571          0565  1              BEGIN
;  572          0566  1              CURRENTXAB [ XAB$L_NXT ] = .NEWXAB;
;  573          0567  1              CURRENTXAB = .NEWXAB;
;  574          0568  1              CURRENTXAB [ XAB$B_COD ] = XAB$C_ALL;
;  575          0569  1              CURRENTXAB [ XAB$B_BLN ] = XAB$C_ALLLEN;
;  576          0570  1              CURRENTXAB [ XAB$B_AID ] = .I;
;  577          0571  1              NEWXAB = .NEWXAB + XAB$C_ALLLEN
;  578          0572  1              END;
;  579          0573  1          !
;  580          0574  1          ! The last xab points to 0
;  581          0575  1          !
;  582          0576  1          CURRENTXAB [ XAB$L_NXT ] = 0;
;  583          0577  1          !
;  584          0578  1          ! Do a display to fill it all in
;  585          0579  1          !
;  586          0580  1          $DISPLAY ( FAB=CONV$AB_IN_FAB )
;  587          0581  1          !
;  588          0582  1          END;
;  589          0583  1          !
;  590          0584  1          ! If this is an indexed file then set the key of ref. to input on
;  591          0585  1          !
;  592          0586  1          IF .CONV$AB_IN_FAB [ FAB$B_ORG ] EQL FAB$C_IDX
;  593          0587  1          THEN
;  594          0588  1              !
;  595          0589  1              ! If the key of ref. is out of range then signal an error and return
;  596          0590  1              ! normal. (so we can continue)
;  597          0591  1              !
;  598          0592  1              IF .CONV$GL_KEY GEQ .CONV$AB_IN_XABSUM [ XAB$B_NOK ]
;  599          0593  1              THEN
;  600          0594  1                  RETURN CONV$_NOKEY
;  601          0595  1              ELSE
;  602          0596  1                  CONV$AB_IN_RAB [ RAB$B_KRF ] = .CONV$GL_KEY;
;  603          0597  1          !
;  604          0598  1          ! Must Special Case for a UDF (Undefined) Input File
;  605          0599  1          !
```

```
;  606     0600  2     IF .CONV$AB_IN_FAB [ FAB$B_RFM ] EQL FAB$C_UDF
;  607     0601  2     THEN
;  608     0602  2         BEGIN
;  609     0603  2
;  610     0604  2         ! Get ready to input the file with Block IO
;  611     0605  2         !
;  612     0606  2         CONV$AB_IN_RAB [ RAB$L_BKT ] = 0;
;  613     0607  2         CONV$AB_IN_RAB [ RAB$V_BIO ] = _SET
;  614     0608  2         END
;  615     0609  2     ELSE
;  616     0610  2
;  617     0611  2         ! Else do normal record IO
;  618     0612  2         !
;  619     0613  2         CONV$AB_IN_RAB [ RAB$V_BIO ] = _CLEAR;
;  620     0614  2
;  621     0615  2     ! In normal operation IN_RAB points to IN_FAB but may be changed
;  622     0616  2     ! when doing sorts
;  623     0617  2     !
;  624     0618  2     CONV$AB_IN_RAB [ RAB$L_FAB ] = CONV$AB_IN_FAB;
;  625     0619  2
;  626     0620  2     ! Now that every thing is ready connect a stream
;  627     0621  2     !
;  628     0622  2     $CONNECT ( RAB=CONV$AB_IN_RAB,ERR=CONV$$RMS_OPEN_ERROR );
;  629     0623  2
;  630     0624  2     ! Any errors from now on are read errors
;  631     0625  2     !
;  632     0626  2     CONV$AB_IN_RAB [ RAB$L_CTX ] = CONV$_READERR;
;  633     0627  2
;  634     0628  2     RETURN CONV$_SUCCESS
;  635     0629  2
;  636     0630  1     END;


                                                    .EXTRN  SYS$OPEN, SYS$DISPLAY
                                                    .EXTRN  SYS$CONNECT

                                    OFFC 00000      .ENTRY  CONV$$OPEN_IN, Save R2,R3,R4,R5,R6,R7,R8,-      ; 0430
                                                            R9,R10,R11
                                57  0000G  CF  9E  00002   MOVAB   CONV$GL_KEY, R7                          ;
                                56  0000G  CF  9E  00007   MOVAB   CONV$AB_IN_XABSUM+9, R6                 ;
                                55  0000G  CF  9E  0000C   MOVAB   CONV$AB_IN_RAB+4, R5                    ;
                                54  0000G  CF  9E  00011   MOVAB   CONV$AB_IN_FAB, R4                      ;
06  A4          01              07  0000G  CF  F0  00016   INSV    CONV$GL_READ_C, #7, #1, CONV$AB_IN_FAB+6  ; 0482
                                08  0000G  CF  E9  0001E   BLBC    CONV$GL_SHARE, 1$                       ; 0486
                        17  A4      4F      8F  90  00023   MOVB    #79, CONV$AB_IN_FAB+23                  ; 0493
                            65              08  88  00028   BISB2   #8, CONV$AB_IN_RAB+4                    ; 0497
                                            67  D5  0002B 1$: TSTL    CONV$GL_KEY                             ; 0505
                                            05  12  0002D   BNEQ    2$                                      ;
                                05  0000G  CF  E9  0002F   BLBC    CONV$GL_SORT, 3$                        ;
                        04  A4      40      8F  8A  00034 2$: BICB2   #64, CONV$AB_IN_FAB+4                  ; 0507
                                0000G  CF      9F  00039 3$: PUSHAB  CONV$$RMS_OPEN_ERROR                    ; 0511
                                            54  DD  0003D   PUSHL   R4                                      ;
                    00000000G   00          02  FB  0003F   CALLS   #2, SYS$OPEN                            ;
                        0000G   CF          01  88  00046   BISB2   #1, CONV$AB_FLAGS+2                     ; 0515
                                20      1D  A4  91  0004B   CMPB    CONV$AB_IN_FAB+29, #32                  ; 0520
                                            7B  12  0004F   BNEQ    8$                                      ;
```

```
                     76     0000G  CF   E9 00051          BLBC    CONV$GL_CREATE, 8$                                      ; 0521
                     71     0000G  CF   E8 00056          BLBS    CONV$GL_FDL, 8$
                     51            66   9A 0005B          MOVZBL  CONV$AB_IN_XABSUM+9, BYTES                              ; 0535
                     51 0000006C   8F   C4 0005E          MULL2   #108, BYTES
                     50       FF   A6   9A 00065          MOVZBL  CONV$AB_IN_XABSUM+8, R0                                 ; 0536
                     50            20   C4 00069          MULL2   #32, R0
                     51            50   C0 0006C          ADDL2   R0, BYTES
                                   51   DD 0006F          PUSHL   BYTES                                                   ; 0540
                                0000G 30   00071          BSBW    CONV$$GET_VM
                     5E            04   C0 00074          ADDL2   #4, SP
                     51       F7   A6   9E 00077          MOVAB   CONV$AB_IN_XABSUM, CURRENTXAB                           ; 0544
                     53            66   9A 0007B          MOVZBL  CONV$AB_IN_XABSUM+9, R3                                 ; 0549
                     52            01   CE 0007E          MNEGL   #1, I
                                   19   11 00081          BRB     5$
               04    A1            50   D0 00083 4$:      MOVL    NEWXAB, 4(CURRENTXAB)                                   ; 0552
                     51            80   7E 00087          MOVAQ   (NEWXAB)+, CURRENTXAB                                   ; 0553
                     61      4C15  8F   B0 0008A          MOVW    #19477, (CURRENTXAB)                                    ; 0554
               17    A1            52   90 0008F          MOVB    I, 23(CURRENTXAB)                                       ; 0556
               38    A1       4C   A1   9E 00093          MOVAB   76(R1), 56(CURRENTXAB)                                  ; 0557
                     50       64   A0   9E 00098          MOVAB   100(R0), NEWXAB                                         ; 0558
          E3         52            53   F2 0009C 5$:      AOBLSS  R3, I, 4$
                     53       FF   A6   9A 000A0          MOVZBL  CONV$AB_IN_XABSUM+8, R3                                 ; 0563
                     52            01   CE 000A4          MNEGL   #1, I
                                   13   11 000A7          BRB     7$
               04    A1            50   D0 000A9 6$:      MOVL    NEWXAB, 4(CURRENTXAB)                                   ; 0566
                     51            80   7E 000AD          MOVAQ   (NEWXAB)+, CURRENTXAB                                   ; 0567
                     61      2014  8F   B0 000B0          MOVW    #8212, (CURRENTXAB)                                     ; 0568
               17    A1            52   90 000B5          MOVB    I, 23(CURRENTXAB)                                       ; 0570
                     50            18   C0 000B9          ADDL2   #24, NEWXAB                                             ; 0571
          E9         52            53   F2 000BC 7$:      AOBLSS  R3, I, 6$
                        04         A1   D4 000C0          CLRL    4(CURRENTXAB)                                           ; 0576
                                   54   DD 000C3          PUSHL   R4                                                      ; 0580
                00000000G  00            01   FB 000C5          CALLS   #1, SYS$DISPLAY
                     20       1D   A4   91 000CC 8$:      CMPB    CONV$AB_IN_FAB+29, #32                                  ; 0586
                                   13   12 000D0          BNEQ    10$
          67         66       08        00   ED 000D2          CMPZV   #0, #8, CONV$AB_IN_XABSUM+9, CONV$GL_KEY         ; 0592
                                   08   14 000D7          BGTR    9$
                     50 00000000G  8F   D0 000D9          MOVL    #CONV$_NOKEY, R0                                        ; 0594
                                   04   00E0             RET
               31    A5            67   90 000E1 9$:      MOVB    CONV$GL_KEY, CONV$AB_IN_RAB+53                          ; 0596
                        1F         A4   95 000E5 10$:     TSTB    CONV$AB_IN_FAB+31                                       ; 0600
                                   09   12 000E8          BNEQ    11$
                        34         A5   D4 000EA          CLRL    CONV$AB_IN_RAB+56                                       ; 0606
               01    A5            08   88 000ED          BISB2   #8, CONV$AB_IN_RAB+5                                    ; 0607
                                   04   11 000F1          BRB     12$
               01    A5            08   8A 000F3 11$:     BICB2   #8, CONV$AB_IN_RAB+5                                    ; 0613
               38    A5            64   9E 000F7 12$:     MOVAB   CONV$AB_IN_FAB, CONV$AB_IN_RAB+60                       ; 0618
                           0000G   CF   9F 000FB          PUSHAB  CONV$$RMS_OPEN_ERROR                                    ; 0622
                              FC   A5   9F 000FF          PUSHAB  CONV$AB_IN_RAB
                00000000G  00            02   FB 00102          CALLS   #2, SYS$CONNECT
                     14    A5 00000000G  8F   D0 00109          MOVL    #CONV$_READERR, CONV$AB_IN_RAB+24               ; 0626
                           50            01   D0 00111          MOVL    #1, R0                                          ; 0628
                                         04   0114             RET                                                     ; 0630

; Routine Size:  277 bytes,    Routine Base:  _CONV$CODE + 00FB
```

```
;   638          0631  1 %SBTTL  'OPEN_OUTPUT'
;   639          0632  1 GLOBAL ROUTINE  CONV$$OPEN_OUTPUT =
;   640          0633  1 !++
;   641          0634  1 !
;   642          0635  1 ! Functional Description:
;   643          0636  1 !
;   644          0637  1 !       Creates ( or opens ) the output file, connects a record stream and if
;   645          0638  1 !       it is an indexed file allocates and fills in the prologue key and
;   646          0639  1 !       area descriptor blocks for sort and/or fast load
;   647          0640  1 !
;   648          0641  1 ! Calling Sequence:
;   649          0642  1 !
;   650          0643  1 !       CONV$$OPEN_OUTPUT
;   651          0644  1 !
;   652          0645  1 ! Input Parameters:
;   653          0646  1 !       none
;   654          0647  1 !
;   655          0648  1 ! Implicit Inputs:
;   656          0649  1 !
;   657          0650  1 !       CONV$AB_OUT_FAB - Output fab
;   658          0651  1 !       CONV$AB_IN_FAB  - Input fab
;   659          0652  1 !       Option flags
;   660          0653  1 !
;   661          0654  1 ! Output Parameters:
;   662          0655  1 !       none
;   663          0656  1 !
;   664          0657  1 ! Implicit Outputs:
;   665          0658  1 !
;   666          0659  1 !       CONV$AB_FLAGS [ CONV$V_OUT ]    - Set on success
;   667          0660  1 !
;   668          0661  1 ! Routine Value:
;   669          0662  1 !
;   670          0663  1 !       CONV$_SUCCESS or error from CONV$$OPEN_IN
;   671          0664  1 !
;   672          0665  1 ! Routines Called:
;   673          0666  1 !
;   674          0667  1 !       $PARSE
;   675          0668  1 !       CONV$$RMS_OPEN_ERROR    - By RMS as an AST
;   676          0669  1 !       $CREATE
;   677          0670  1 !       $DISPLAY
;   678          0671  1 !       $OPEN
;   679          0672  1 !       CONV$$READ_PROLOGUE
;   680          0673  1 !       $CONNECT
;   681          0674  1 !       CONV$$OPEN_IN
;   682          0675  1 !
;   683          0676  1 ! Side Effects:
;   684          0677  1 !       none
;   685          0678  1 !
;   686          0679  1 !--
;   687          0680  1
;   688          0681  2     BEGIN
;   689          0682  2
;   690          0683  2     LOCAL
;   691          0684  2         PRESENT : LONG,
;   692          0685  2         OUT_DEV : BLOCK [ 1,LONG ],
;   693          0686  2         STATUS  : LONG;
;   694          0687  2
```

```
;  695     0688  2      ! Any rms errors on the output fab are OPENOUT errors
;  696     0689  2      !
;  697     0690  2      CONV$AB_OUT_FAB [ FAB$L_CTX ] = CONV$_OPENOUT;
;  698     0691  2
;  699     0692  2      ! Use the file name in the call argument (not one from FDL)
;  700     0693  2      !
;  701     0694  2      CONV$AB_OUT_FAB [ FAB$B_FNS ] = .CONV$AR_OUT_FILE_NAM [ DSC$W_LENGTH ];
;  702     0695  2      CONV$AB_OUT_FAB [ FAB$L_FNA ] = .CONV$AR_OUT_FILE_NAM [ DSC$A_POINTER ];
;  703     0696  2
;  704     0697  2      ! Setup the input file name for output file related file parsing
;  705     0698  2      !
;  706     0699  2      CONV$AB_IN_NAM [ NAM$B_RSL ] = .CONV$AB_IN_FAB [ FAB$B_FNS ];
;  707     0700  2      CONV$AB_IN_NAM [ NAM$L_RSA ] = .CONV$AB_IN_FAB [ FAB$L_FNA ];
;  708     0701  2
;  709     0702  2      ! Parse the name
;  710     0703  2      !
;  711     0704  2      $PARSE ( FAB=CONV$AB_OUT_FAB,ERR=CONV$$RMS_OPEN_ERROR );
;  712     0705  2
;  713     0706  2      ! Check the device that the output file is on
;  714     0707  2      !
;  715     0708  2      OUT_DEV = .CONV$AB_OUT_FAB [ FAB$L_DEV ];
;  716     0709  2
;  717     0710  2      ! We cannot fast load to a network device
;  718     0711  2      !
;  719     0712  2      IF .OUT_DEV [ DEV$V_NET ]
;  720     0713  2      THEN
;  721     0714  2          CONV$GL_FAST = _CLEAR;
;  722     0715  2
;  723     0716  2      ! Set the FAB from the Option Switches
;  724     0717  2      !
;  725     0718  2      ! Write Check
;  726     0719  2      !
;  727     0720  2      CONV$AB_OUT_FAB [ FAB$V_WCK ] = .CONV$GL_WRITE_C;
;  728     0721  2
;  729     0722  2      ! Merge
;  730     0723  2      !
;  731     0724  2      CONV$AB_OUT_FAB [ FAB$V_SQO ] = ( NOT .CONV$GL_MERGE );
;  732     0725  2
;  733     0726  2      ! If the CREATE Option was specified then Create the output file
;  734     0727  2      ! else just open it
;  735     0728  2      !
;  736     0729  2      IF .CONV$GL_CREATE
;  737     0730  2      THEN
;  738     0731  3          BEGIN
;  739     0732  3
;  740     0733  3          LOCAL COPY_FAB  : REF BLOCK [ ,BYTE ];
;  741     0734  3
;  742     0735  3          ! Determine where to copy fab from
;  743     0736  3          !
;  744     0737  3          IF .CONV$GL_FDL
;  745     0738  3          THEN
;  746     0739  4              BEGIN
;  747     0740  4
;  748     0741  4              ! If fdl was done copy the stuff from the fab produced by
;  749     0742  4              ! fdl$$parse
;  750     0743  4              !
;  751     0744  4              COPY_FAB = .CONV$AB_FDL_FAB;
```

```
;  752          0745  4
;  753          0746  4                ! Connect the fdl xabs
;  754          0747  4                !
;  755          0748  4                CONV$AB_OUT_XABSUM [ XAB$L_NXT ] = .COPY_FAB [ FAB$L_XAB ]
;  756          0749  4
;  757          0750  4                END
;  758          0751  3            ELSE
;  759          0752  4                BEGIN
;  760          0753  4
;  761          0754  4                ! If this is not a create by FDL definition then get the stuff
;  762          0755  4                ! from the input file
;  763          0756  4                !
;  764          0757  4                COPY_FAB = CONV$AB_IN_FAB;
;  765          0758  4
;  766          0759  4                ! Connect the input files summary xab NXT which will connect
;  767          0760  4                ! any other xabs that the input file may have had ie. area and
;  768          0761  4                ! key xabs
;  769          0762  4                !
;  770          0763  4                CONV$AB_OUT_XABSUM [ XAB$L_NXT ] = .CONV$AB_IN_XABSUM [ XAB$L_NXT ]
;  771          0764  4
;  772          0765  4                END;
;  773          0766  3
;  774          0767  3            ! Copy the important fab fields
;  775          0768  3            !
;  776          0769  3            CONV$AB_OUT_FAB [ FAB$L_ALQ ] = .COPY_FAB [ FAB$L_ALQ ]; ! Allocation
;  777          0770  3            CONV$AB_OUT_FAB [ FAB$W_DEQ ] = .COPY_FAB [ FAB$W_DEQ ]; ! Extension
;  778          0771  3            CONV$AB_OUT_FAB [ FAB$B_RTV ] = .COPY_FAB [ FAB$B_RTV ]; ! Reteval windoow
;  779          0772  3            CONV$AB_OUT_FAB [ FAB$B_ORG ] = .COPY_FAB [ FAB$B_ORG ]; ! Organization
;  780          0773  3            CONV$AB_OUT_FAB [ FAB$B_RAT ] = .COPY_FAB [ FAB$B_RAT ]; ! Record attributes
;  781          0774  3            CONV$AB_OUT_FAB [ FAB$B_RFM ] = .COPY_FAB [ FAB$B_RFM ]; ! Record format
;  782          0775  3            CONV$AB_OUT_FAB [ FAB$W_MRS ] = .COPY_FAB [ FAB$W_MRS ]; ! Max record size
;  783          0776  3            CONV$AB_OUT_FAB [ FAB$L_MRN ] = .COPY_FAB [ FAB$L_MRN ]; ! Max records
;  784          0777  3            CONV$AB_OUT_FAB [ FAB$W_BLS ] = .COPY_FAB [ FAB$W_BLS ]; ! Block size
;  785          0778  3            CONV$AB_OUT_FAB [ FAB$B_BKS ] = .COPY_FAB [ FAB$B_BKS ]; ! Bucket size
;  786          0779  3            CONV$AB_OUT_FAB [ FAB$B_FSZ ] = .COPY_FAB [ FAB$B_FSZ ]; ! Fixed size
;  787          0780  3            CONV$AB_OUT_FAB [ FAB$W_GBC ] = .COPY_FAB [ FAB$W_GBC ]; ! Global Buffers
;  788          0781  3
;  789          0782  3            CONV$AB_OUT_FAB [ FAB$L_FOP ] = .CONV$AB_OUT_FAB [ FAB$L_FOP ] OR
;  790          0783  3                                            .COPY_FAB [ FAB$L_FOP ]; ! File options
;  791          0784  3
;  792          0785  3            ! If the PROLOGUE option was specified and the file is indexed
;  793          0786  3            ! then stuff the first key xab with the user value
;  794          0787  3            !
;  795          0788  3            IF ( .CONV$AB_OUT_FAB [ FAB$B_ORG ] EQLU FAB$C_IDX ) AND
;  796          0789  3                                            .CONV$AB_FLAGS [ CONV$V_PROLOG ]
;  797          0790  3            THEN
;  798          0791  4                BEGIN
;  799          0792  4
;  800          0793  4                LOCAL       XAB : REF BLOCK [ ,BYTE ];
;  801          0794  4
;  802          0795  4                ! Find the first key xab
;  803          0796  4                !
;  804          0797  4                XAB = .CONV$AB_OUT_FAB [ FAB$L_XAB ];
;  805          0798  4
;  806          0799  4                ! The xabs have to be in order and there must be a key 0 so
;  807          0800  4                ! the first one we find is the one we want
;  808          0801  4                !
```

```
;  809          0802  4          WHILE .XAB [ XAB$B_COD ] NEQU XAB$C_KEY
;  810          0803  4          DO
;  811          0804  4
;  812          0805  4              ! If there are no more xabs then we really have a problem
;  813          0806  4              ! so forget it
;  814          0807  4              !
;  815          0808  4              IF .XAB [ XAB$L_NXT ] EQLU 0
;  816          0809  4              THEN
;  817          0810  4                  RETURN CONV$_BADLOGIC
;  818          0811  4              ELSE
;  819          0812  4                  XAB = .XAB [ XAB$L_NXT ];
;  820          0813  4              !
;  821          0814  4              ! Stuff the value
;  822          0815  4              !
;  823          0816  4              XAB [ XAB$B_PROLOG ] = .CONV$GL_PROLOG
;  824          0817  4              !
;  825          0818  4              END;
;  826          0819
;  827          0820                 ! Create it
;  828          0821                 !
;  829          0822                 ! If the record format was changed on a non VMS system
;  830          0823                 ! signal a warning (only to DCL)
;  831          0824                 !
;  832          0825                 $CREATE ( FAB=CONV$AB_OUT_FAB,ERR=CONV$$RMS_OPEN_ERROR );
;  833          0826
;  834          0827    !            IF ( $CREATE ( FAB=CONV$AB_OUT_FAB,ERR=CONV$$RMS_OPEN_ERROR ) EQLU
;  835          0828    !                                                           RMS$_CRE_STM ) AND
;  836          0829    !                                                  .CONV$AB_FLAGS [ CONV$V_DCL ]
;  837          0830                 THEN
;  838          0831    !                SIGNAL ( CONV$_CREATEDSTM );
;  839          0832    !
;  840          0833                 ! Since a create does not fill in the summary xab do a display
;  841          0834                 !
;  842          0835  4              $DISPLAY( FAB=CONV$AB_OUT_FAB )
;  843          0836  4
;  844          0837                 END
;  845          0838                 ELSE
;  846          0839                     $OPEN ( FAB=CONV$AB_OUT_FAB,ERR=CONV$$RMS_OPEN_ERROR );
;  847          0840
;  848          0841                 ! If we got here then we have opened a file.
;  849          0842                 !
;  850          0843                 CONV$AB_FLAGS [ CONV$V_OUT ] = _SET;
;  851          0844
;  852          0845                 ! Set some bits depending on the type of output file
;  853          0846                 !
;  854          0847                 ! Can only append to a sequential file
;  855          0848                 !
;  856          0849                 IF .CONV$AB_OUT_FAB [ FAB$B_ORG ] EQLU FAB$C_SEQ
;  857          0850                 THEN
;  858          0851                     CONV$AB_OUT_RAB [ RAB$V_EOF ] = .CONV$GL_APPEND
;  859          0852                 ELSE
;  860          0853
;  861          0854                     ! If append was on without a seq. output file then error
;  862          0855                     !
;  863          0856                     IF .CONV$GL_APPEND THEN RETURN CONV$_NOTSEQ;
;  864          0857
;  865          0858  2              ! Is't not very exciting if it's not an index file
```

```
;   866     0859  2     !
;   867     0860  2     IF .CONV$AB_OUT_FAB [ FAB$B_ORG ] NEQU FAB$C_IDX
;   868     0861  2     THEN
;   869     0862  2         BEGIN
;   870     0863  2         CONV$GL_MERGE = _CLEAR;
;   871     0864  2         CONV$GL_SORT  = _CLEAR;
;   872     0865  2         CONV$GL_FAST  = _CLEAR
;   873     0866  2         END
;   874     0867  2     ELSE
;   875     0868  2
;   876     0869  2         ! Set the fill option if it is indexed
;   877     0870  2         !
;   878     0871  2         CONV$AB_OUT_RAB [ RAB$V_LOA ] = NOT .CONV$GL_FILL;
;   879     0872  2
;   880     0873  2     ! If we are sorting or fastloading
;   881     0874  2     ! then allocate space for KEY and AREA XAB's and fill them in by reading
;   882     0875  2     ! the prologue blocks in the file
;   883     0876  2     !
;   884     0877  2     IF ( .CONV$GL_FAST OR .CONV$GL_SORT )
;   885     0878  2     THEN
;   886     0879  2         BEGIN
;   887     0880  2
;   888     0881  2         ! Connect the file for Block IO for reading the
;   889     0882  2         ! prologue.
;   890     0883  2         !
;   891     0884  2         CONV$AB_OUT_RAB [ RAB$V_BIO ] = _SET;
;   892     0885  2
;   893     0886  2         $CONNECT ( RAB=CONV$AB_OUT_RAB,ERR=CONV$$RMS_OPEN_ERROR );
;   894     0887  2
;   895     0888  2         ! Read the prologue
;   896     0889  2         !
;   897     0890  2         CONV$$READ_PROLOGUE();
;   898     0891  2
;   899     0892  2         ! If this is not a fast load then we need to bounce the file so we can
;   900     0893  2         ! do record IO again. (This sure doesen'd look good!)
;   901     0894  2         !
;   902     0895  2         IF NOT .CONV$GL_FAST
;   903     0896  3         THEN
;   904     0897  4             BEGIN
;   905     0898  4
;   906     0899  4             ! Disconnect and Close (Dont check the disconnect)
;   907     0900  4             !
;   908     0901  4             $DISCONNECT ( RAB=CONV$AB_OUT_RAB );
;   909     0902  4             $CLOSE( FAB=CONV$AB_OUT_FAB );
;   910     0903  4
;   911     0904  4             ! Clear the Block IO flag
;   912     0905  4             !
;   913     0906  4             CONV$AB_OUT_RAB [ RAB$V_BIO ] = _CLEAR;
;   914     0907  4
;   915     0908  4             ! Reopen and Reconnect (Dont need to reconnect the PLG RAB)
;   916     0909  4             !
;   917     0910  4             $OPEN ( FAB=CONV$AB_OUT_FAB,ERR=CONV$$RMS_OPEN_ERROR );
;   918     0911  4
;   919     0912  5             $CONNECT ( RAB=CONV$AB_OUT_RAB,ERR=CONV$$RMS_OPEN_ERROR )
;   920     0913  5
;   921     0914  4             END
;   922     0915  3         END
```

```
;  923          0916  2      ELSE
;  924          0917  2          BEGIN
;  925          0918  2
;  926          0919  2          ! If we are merging into an indexed file
;  927          0920  2          ! then set the access to KEY
;  928          0921  2          !
;  929          0922  2          IF .CONV$GL_MERGE
;  930          0923  2          THEN
;  931          0924  2              CONV$AB_OUT_RAB [ RAB$B_RAC ] = RAB$C_KEY;
;  932          0925  2
;  933          0926  2          ! If we are not sorting or fastloading
;  934          0927  2          ! then connect the stream normally
;  935          0928  2          !
;  936          0929  2          $CONNECT ( RAB=CONV$AB_OUT_RAB,ERR=CONV$$RMS_OPEN_ERROR );
;  937          0930  2
;  938          0931  2          ! If the output file was not opened by now we can open it here
;  939          0932  2          !
;  940          0933  2          IF NOT .CONV$AB_FLAGS [ CONV$V_IN ]
;  941          0934  2          THEN
;  942          0935  3              RET_ON_ERROR( CONV$$OPEN_IN() )
;  943          0936  3
;  944          0937  2          END;
;  945          0938  2
;  946          0939  2      ! If PAD switch is on and the file is not fixed format
;  947          0940  2      !
;  948          0941  2      IF .CONV$GL_PAD AND ( .CONV$AB_OUT_FAB [ FAB$B_RFM ] NEQU FAB$C_FIX )
;  949          0942  2      THEN
;  950          0943  2          BEGIN
;  951          0944  2          CONV$GL_PAD = _CLEAR;
;  952          0945  2          SIGNAL( CONV$_PAD )
;  953          0946  2          END;
;  954          0947  2
;  955          0948  2      ! Any errors on the output rab should be write errors (exceptions are in
;  956          0949  2      ! the fast load code
;  957          0950  2      !
;  958          0951  2      CONV$AB_OUT_RAB [ RAB$L_CTX ] = CONV$_WRITEERR;
;  959          0952  2
;  960          0953  2      ! Return normally
;  961          0954  2      !
;  962          0955  2      RETURN CONV$_SUCCESS
;  963          0956  2
;  964          0957  1      END;


                                                .EXTRN  SYS$PARSE, SYS$CREATE
                                                .EXTRN  SYS$DISCONNECT, SYS$CLOSE

                                 0FFC 00000     .ENTRY  CONV$$OPEN_OUTPUT, Save R2,R3,R4,R5,R6,R7,- ; 0632
                                                        R8,R9,R10,R11
                    59      0000G  CF  9E 00002 MOVAB   CONV$AB_FLAGS+2, R9                         ;
                    58      0000G  CF  9E 00007 MOVAB   CONV$GL_MERGE, R8                           ;
                    57 00000000G   00  9E 0000C MOVAB   SYS$OPEN, R7                                ;
                    56      0000G  CF  9E 00013 MOVAB   CONV$GL_FAST, R6                            ;
                    55 00000000G   00  9E 00018 MOVAB   SYS$CONNECT, R5                             ;
                    54      0000G  CF  9E 0001F MOVAB   CONV$$RMS_OPEN_ERROR, R4                    ;
                    53      0000G  CF  9E 00024 MOVAB   CONV$AB_OUT_RAB+4, R3                       ;
```

```
                                52     0000G   CF   9E 00029          MOVAB   CONV$AB_OUT_FAB, R2                                    ;0690
                             18 A2  00000000G   8F   D0 0002E          MOVL    #CONV$_OPENOUT, CONV$AB_OUT_FAB+24                    ;0694
                                50     0000G   CF   D0 00036          MOVL    CONV$AR_OUT_FILE_NAM, R0
                             34 A2             60   90 0003B          MOVB    (R0), CONV$AB_OUT_FAB+52                              ;0695
                             2C A2         04  A0   D0 0003F          MOVL    4(R0), CONV$AB_OUT_FAB+44                             ;0699
                          0000G CF      0000G  CF   90 00044          MOVB    CONV$AB_IN_FAB+52, CONV$AB_IN_NAM+3                   ;0700
                          0000G CF      0000G  CF   D0 0004B          MOVL    CONV$AB_IN_FAB+44, CONV$AB_IN_NAM+4                   ;0704
                                              14   BB 00052          PUSHR   #^M<R2,R4>
                      00000000G 00            02   FB 00054          CALLS   #2, SYS$PARSE                                         ;0708
                                50         40  A2   D0 0005B          MOVL    CONV$AB_OUT_FAB+64, OUT_DEV                           ;0712
                          02            50        0D   E1 0005F          BBC     #13, OUT_DEV, 1$                                      ;0714
                                              66   D4 00063          CLRL    CONV$GL_FAST                                          ;0720
05 A2                     01            01     0000G  CF   F0 00065 1$:    INSV    CONV$GL_WRITE_C, #1, #1, CONV$AB_OUT_FAB+5            ;0724
                                50         68   D2 0006D          MCOML   CONV$GL_MERGE, R0
04 A2                     01            06        50   F0 00070          INSV    R0, #6, #1, CONV$AB_OUT_FAB+4                         ;0729
                                03     0000G  CF   E8 00076          BLBS    CONV$GL_CREATE, 2$
                                          0081  31   0007B          BRW     9$                                                    ;0737
                                0D     0000G  CF   E9 0007E 2$:     BLBC    CONV$GL_FDL, 3$                                       ;0744
                                50     0000'  CF   D0 00083          MOVL    CONV$AB_FDL_FAB, COPY_FAB                             ;0748
                          0000G CF        24  A0   D0 00088          MOVL    36(COPY_FAB), CONV$AB_OUT_XABSUM+4                    ;0757
                                              0C   11 0008E          BRB     4$
                                50     0000G  CF   9E 00090 3$:     MOVAB   CONV$AB_IN_FAB, COPY_FAB                              ;0763
                          0000G CF      0000G  CF   D0 00095          MOVL    CONV$AB_IN_XABSUM+4, CONV$AB_OUT_XABSUM+4             ;0769
                             10 A2         10  A0   D0 0009C 4$:     MOVL    16(COPY_FAB), CONV$AB_OUT_FAB+16                      ;0770
                             14 A2         14  A0   B0 000A1          MOVW    20(COPY_FAB), CONV$AB_OUT_FAB+20                      ;0771
                             1C A2         1C  A0   D0 000A6          MOVL    28(COPY_FAB), CONV$AB_OUT_FAB+28                      ;0775
                             36 A2         36  A0   B0 000AB          MOVW    54(COPY_FAB), CONV$AB_OUT_FAB+54                      ;0776
                             38 A2         38  A0   7D 000B0          MOVQ    56(COPY_FAB), CONV$AB_OUT_FAB+56                      ;0780
                             48 A2         48  A0   B0 000B5          MOVW    72(COPY_FAB), CONV$AB_OUT_FAB+72                      ;0783
                             04 A2         04  A0   C8 000BA          BISL2   4(COPY_FAB), CONV$AB_OUT_FAB+4                        ;0788
                                20         1D  A2   91 000BF          CMPB    CONV$AB_OUT_FAB+29, #32                               ;0789
                                              26   12 000C3          BNEQ    8$
                          22            69        06   E1 000C5          BBC     #6, CONV$AB_FLAGS+2, 8$                               ;0797
                                50         24  A2   D0 000C9          MOVL    CONV$AB_OUT_FAB+36, XAB                               ;0802
                                15            60   91 000CD 5$:     CMPB    (XAB), #21
                                              13   13 000D0          BEQL    7$                                                    ;0808
                                           04  A0   D5 000D2          TSTL    4(XAB)
                                              08   12 000D5          BNEQ    6$                                                    ;0810
                                50 00000000G   8F   D0 000D7          MOVL    #CONV$_BADLOGIC, R0
                                              04   000DE          RET                                                           ;0812
                                50         04  A0   D0 000DF 6$:     MOVL    4(XAB), XAB                                           ;0808
                                              E8   11 000E3          BRB     5$                                                    ;0816
                             48 A0      0000G  CF   90 000E5 7$:     MOVB    CONV$GL_PROLOG, 72(XAB)                               ;0825
                                              14   BB 000EB 8$:     PUSHR   #^M<R2,R4>
                      00000000G 00            02   FB 000ED          CALLS   #2, SYS$CREATE                                        ;0835
                                              52   DD 000F4          PUSHL   R2
                      00000000G 00            01   FB 000F6          CALLS   #1, SYS$DISPLAY
                                              05   11 000FD          BRB     10$                                                   ;0839
                                              14   BB 000FF 9$:     PUSHR   #^M<R2,R4>
                                67            02   FB 00101          CALLS   #2, SYS$OPEN                                          ;0843
                                69            02   88 00104 10$:    BISB2   #2, CONV$AB_FLAGS+2                                   ;0849
                                50         1D  A2   9A 00107          MOVZBL  CONV$AB_OUT_FAB+29, R0
                                              0A   12 0010B          BNEQ    11$                                                   ;0851
01 A3                     01            00     0000G  CF   F0 0010D          INSV    CONV$GL_APPEND, #0, #1, CONV$AB_OUT_RAB+5             ;0856
                                              0D   11 00115          BRB     12$
                                08     0000G  CF   E9 00117 11$:     BLBC    CONV$GL_APPEND, 12$
                                50 00000000G   8F   D0 0011C          MOVL    #CONV$_NOTSEQ, R0
```

```
                                                 04 00123          RET
                                20               50  91 00124 12$:  CMPB    R0, #32                                          ;  0860
                                                 0A  13 00127       BEQL    13$
                                                 68  D4 00129       CLRL    CONV$GL_MERGE                                    ;  0863
                                          0000G  CF  D4 0012B       CLRL    CONV$GL_SORT                                     ;  0864
                                                 66  D4 0012F       CLRL    CONV$GL_FAST                                     ;  0865
                                                 0B  11 00131       BRB     14$
                                50       0000G  CF  D2 00133 13$:  MCOML   CONV$GL_FILL, R0                                 ;  0871
01  A3                  01      05               50  F0 00138       INSV    R0, #5, #1, CONV$AB_OUT_RAB+5
                                05               66  E8 0013E 14$:  BLBS    CONV$GL_FAST, 15$                                ;  0877
                                38       0000G  CF  E9 00141       BLBC    CONV$GL_SORT, 16$
                          01    A3               08  88 00146 15$:  BISB2   #8, CONV$AB_OUT_RAB+5                            ;  0884
                                                 54  DD 0014A       PUSHL   R4                                               ;  0886
                                   FC    A3      9F  9F 0014C       PUSHAB  CONV$AB_OUT_RAB
                                65               02  FB 0014F       CALLS   #2, SYS$CONNECT
                                          0000G  30  00152          BSBW    CONV$$READ_PROLOGUE                              ;  0890
                                40               66  E8 00155       BLBS    CONV$GL_FAST, 18$                                ;  0895
                                   FC    A3      9F  9F 00158       PUSHAB  CONV$AB_OUT_RAB                                  ;  0901
                    00000000G   00               01  FB 0015B       CALLS   #1, SYS$DISCONNECT
                                                 52  DD 00162       PUSHL   R2                                               ;  0902
                    00000000G   00               01  FB 00164       CALLS   #1, SYS$CLOSE
                          01    A3               08  8A 0016B       BICB2   #8, CONV$AB_OUT_RAB+5                            ;  0906
                                                 14  BB 0016F       PUSHR   #^M<R2,R4>                                       ;  0910
                                67               02  FB 00171       CALLS   #2, SYS$OPEN
                                                 54  DD 00174       PUSHL   R4                                               ;  0912
                                   FC    A3      9F  9F 00176       PUSHAB  CONV$AB_OUT_RAB
                                65               02  FB 00179       CALLS   #2, SYS$CONNECT
                                                 1A  11 0017C       BRB     18$                                              ;  0895
                                04               68  E9 0017E 16$:  BLBC    CONV$GL_MERGE, 17$                               ;  0922
                             1A    A3               01  90 00181       MOVB    #1, CONV$AB_OUT_RAB+30                           ;  0924
                                                 54  DD 00185 17$:  PUSHL   R4                                               ;  0929
                                   FC    A3      9F  9F 00187       PUSHAB  CONV$AB_OUT_RAB
                                65               02  FB 0018A       CALLS   #2, SYS$CONNECT
                                08               69  E8 0018D       BLBS    CONV$AB_FLAGS+2, 18$                             ;  0933
                          FD56  CF               00  FB 00190       CALLS   #0, CONV$$OPEN_IN                                ;  0935
                                27               50  E9 00195       BLBC    STATUS, 20$
                                17       0000G  CF  E9 00198 18$:  BLBC    CONV$GL_PAD, 19$                                 ;  0941
                                01         1F   A2  91 0019D       CMPB    CONV$AB_OUT_FAB+31, #1
                                                 11  13 001A1       BEQL    19$
                                         0000G  CF  D4 001A3       CLRL    CONV$GL_PAD                                      ;  0944
                                   00000000G     8F  DD 001A7       PUSHL   #CONV$_PAD                                       ;  0945
                    00000000G   00               01  FB 001AD       CALLS   #1, LIB$SIGNAL
                          14    A3   00000000G     8F  D0 001B4 19$:  MOVL    #CONV$_WRITEERR, CONV$AB_OUT_RAB+24              ;  0951
                                50               01  D0 001BC       MOVL    #1, R0                                           ;  0955
                                                 04  001BF 20$:  RET                                                         ;  0957

; Routine Size:  448 bytes,    Routine Base:  _CONV$CODE + 0210
```

```
;  966     0958  1 %SBTTL  'CREATE_BUFFER'
;  967     0959  1 GLOBAL ROUTINE  CONV$$CREATE_BUFFER =
;  968     0960  1 !++
;  969     0961  1 !
;  970     0962  1 ! Functional Description:
;  971     0963  1 !
;  972     0964  1 !       Creates the main record buffer and sets the record sizes
;  973     0965  1 !
;  974     0966  1 !       The main record buffer:
;  975     0967  1 !
;  976     0968  1 !               ----------------------------------------//----------------
;  977     0969  1 !               |                |                                      |
;  978     0970  1 !               |                |                                      |
;  979     0971  1 !               ----------------------------------------//----------------
;  980     0972  1 !               ^                ^
;  981     0973  1 !               |                |
;  982     0974  1 !          VFC_BUF_PTR      REC_BUF_PTR
;  983     0975  1 !
;  984     0976  1 !
;  985     0977  1 ! Calling Sequence:
;  986     0978  1 !
;  987     0979  1 !       CONV$$CREATE_BUFFER()
;  988     0980  1 !
;  989     0981  1 ! Input Parameters:
;  990     0982  1 !       none
;  991     0983  1 !
;  992     0984  1 ! Implicit Inputs:
;  993     0985  1 !       none
;  994     0986  1 !
;  995     0987  1 ! Output Parameters:
;  996     0988  1 !       none
;  997     0989  1 !
;  998     0990  1 ! Implicit Outputs:
;  999     0991  1 !
; 1000     0992  1 !       Record buffer pointers and size variables
; 1001     0993  1 !
; 1002     0994  1 ! Routine Value:
; 1003     0995  1 !
; 1004     0996  1 !       SS$_NORMAL
; 1005     0997  1 !
; 1006     0998  1 ! Routines Called:
; 1007     0999  1 !
; 1008     1000  1 !       CONV$$GET_VM
; 1009     1001  1 !
; 1010     1002  1 ! Side Effects:
; 1011     1003  1 !       none
; 1012     1004  1 !
; 1013     1005  1 !--
; 1014     1006  1
; 1015     1007  2     BEGIN
; 1016     1008  2
; 1017     1009  2     LITERAL
; 1018     1010  2         MAX_REC_CTRL     = 14;    ! Maximun number of control bytes for a data
; 1019     1011  2                                   ! record in index file (14 is for a fully
; 1020     1012  2                                   ! compressed prologue 3 record
; 1021     1013  2
; 1022     1014  2     LOCAL
```

```
;  1023     1015  2          IN_VFC,
;  1024     1016  2          IN_MRS,
;  1025     1017  2          OUT_VFC,
;  1026     1018  2          OUT_EXTRA;
;  1027     1019  2
;  1028     1020  2      ! Acccount for the VFC temporaraly
;  1029     1021  2      !
;  1030     1022  2      IF .CONV$AB_OUT_FAB [ FAB$B_RFM ] EQL FAB$C_VFC
;  1031     1023  2      THEN
;  1032     1024  2          OUT_VFC = .CONV$AB_OUT_FAB [ FAB$B_FSZ ]
;  1033     1025  2      ELSE
;  1034     1026  2          OUT_VFC = 0;
;  1035     1027  2
;  1036     1028  2      ! If output MRS = 0 ( ie. VAR and VFC records ) then
;  1037     1029  2      ! check for Block Spanning with Sequential Files
;  1038     1030  2      ! and Bucket Crossing with Relative and Indexed
;  1039     1031  2      !
;  1040     1032  2      IF ( CONV$GW_OUT_MRS = .CONV$AB_OUT_FAB [ FAB$W_MRS ] ) EQL 0
;  1041     1033  2      THEN
;  1042     1034  3          BEGIN
;  1043     1035  3
;  1044     1036  3          LOCAL   OUT_DEV : BLOCK [ 1,LONG ];
;  1045     1037  3
;  1046     1038  3          ! Find out if this thing is going to tape, if so use block size
;  1047     1039  3          ! (Since records cannot spand blocks on tape)
;  1048     1040  3          !
;  1049     1041  3          OUT_DEV = .CONV$AB_OUT_FAB [ FAB$L_DEV ];
;  1050     1042  3
;  1051     1043  3          IF .OUT_DEV [ DEV$V_SQD ]
;  1052     1044  3          THEN
;  1053     1045  3              CONV$GW_OUT_MRS = .CONV$AB_OUT_FAB [ FAB$W_BLS ] - .OUT_VFC - 2
;  1054     1046  3
;  1055     1047  3          ! Sequential and  NO Block spanning
;  1056     1048  3          !
;  1057     1049  3          ELSE IF ( .CONV$AB_OUT_FAB [ FAB$B_ORG ] EQLU FAB$C_SEQ ) AND
;  1058     1050  3                                                  .CONV$AB_OUT_FAB [ FAB$V_BLK ]
;  1059     1051  3          THEN
;  1060     1052  3              CONV$GW_OUT_MRS = BLOCK_SIZE - .OUT_VFC - 2
;  1061     1053  3
;  1062     1054  3          ! Relative
;  1063     1055  3          !
;  1064     1056  3          ELSE IF .CONV$AB_OUT_FAB [ FAB$B_ORG ] EQLU FAB$C_REL
;  1065     1057  3          THEN
;  1066     1058  3              CONV$GW_OUT_MRS = ( .CONV$AB_OUT_FAB [ FAB$B_BKS ] * BLOCK_SIZE ) -
;  1067     1059  3                                                                   .OUT_VFC - 3
;  1068     1060  3
;  1069     1061  3          ! Indexed
;  1070     1062  3          !
;  1071     1063  3          ELSE
;  1072     1064  3              CONV$GW_OUT_MRS = ( .CONV$AB_OUT_FAB [ FAB$B_BKS ] * BLOCK_SIZE ) -
;  1073     1065  3                                                                   .OUT_VFC - 7;
;  1074     1066  3
;  1075     1067  3          END;
;  1076     1068  2
;  1077     1069  2      ! If the Input File is UDF then the UDF_MRS is calculated from
;  1078     1070  2      ! the output file attributes
;  1079     1071  2      !
```

```
; 1080          1072  2     IF .CONV$AB_IN_FAB [ FAB$B_RFM ] EQLU FAB$C_UDF
; 1081          1073  2     THEN
; 1082          1074  2         BEGIN
; 1083          1075  2
; 1084          1076  2         IN_MRS = BLOCK_SIZE;
; 1085          1077  2
; 1086          1078  2         ! If fixed format then no problem use that value, if
; 1087          1079  2         ! not see if a 512 byte record will fit
; 1088          1080  2         !
; 1089          1081  2         IF .CONV$AB_OUT_FAB [ FAB$B_RFM ] EQL FAB$C_FIX
; 1090          1082  2         THEN
; 1091          1083  2             CONV$GW_UDF_MRS = .CONV$AB_OUT_FAB [ FAB$W_MRS ]
; 1092          1084  2         ELSE
; 1093          1085  2
; 1094          1086  2             ! If the udf record will not if into the output file then error
; 1095          1087  2             !
; 1096          1088  2             IF .CONV$GW_OUT_MRS LSS BLOCK_SIZE
; 1097          1089  2             THEN
; 1098          1090  2                 RETURN CONV$_UDF_BLK
; 1099          1091  2             ELSE
; 1100          1092  2                 CONV$GW_UDF_MRS = BLOCK_SIZE
; 1101          1093  2         END
; 1102          1094  2     ELSE
; 1103          1095  2         BEGIN
; 1104          1096  2
; 1105          1097  2         ! Here for a normal input file
; 1106          1098  2         ! IN_MRS is the length of the maximun record size
; 1107          1099  2         !
; 1108          1100  2         ! Now see if the file is VFC
; 1109          1101  2         !
; 1110          1102  2         IF .CONV$AB_IN_FAB [ FAB$B_RFM ] EQL FAB$C_VFC
; 1111          1103  2         THEN
; 1112          1104  2             IN_VFC = .CONV$AB_IN_FAB [ FAB$B_FSZ ]
; 1113          1105  2         ELSE
; 1114          1106  2             IN_VFC = 0;
; 1115          1107  2
; 1116          1108  2         ! If max. record size is zero then we find out from Longest Record Length
; 1117          1109  2         ! on disk or Block Size for magtape
; 1118          1110  2         !
; 1119          1111  2         IF ( IN_MRS = .CONV$AB_IN_FAB [ FAB$W_MRS ] ) EQL 0
; 1120          1112  2         THEN
; 1121          1113  3             BEGIN
; 1122          1114  3
; 1123          1115  3             LOCAL     IN_DEV : BLOCK [ 1,LONG ];
; 1124          1116  3
; 1125          1117  3             ! Find out if this thing is comming from tape if so use block size
; 1126          1118  3             ! (Since records cannot spand blocks on tape)
; 1127          1119  3             !
; 1128          1120  3             IN_DEV = .CONV$AB_IN_FAB [ FAB$L_DEV ];
; 1129          1121  3
; 1130          1122  3             IF .IN_DEV [ DEV$V_SQD ]
; 1131          1123  3             THEN
; 1132          1124  3                 IN_MRS = .CONV$AB_IN_FAB [ FAB$W_BLS ] - .IN_VFC
; 1133          1125  3
; 1134          1126  3             ! If SEQ use LRL otherwise check
; 1135          1127  3             ! bucket sizes
; 1136          1128  3             !
```

```
;  1137     1129  4              ELSE IF .CONV$AB_IN_FAB [ FAB$B_ORG ] EQL FAB$C_SEQ
;  1138     1130  4              THEN
;  1139     1131  4                  IN_MRS = .CONV$AB_IN_XABFHC [ XAB$W_LRL ]
;  1140     1132  4
;  1141     1133  4              ! Relative
;  1142     1134  4              !
;  1143     1135  4              ELSE IF .CONV$AB_IN_FAB [ FAB$B_ORG ] EQL FAB$C_REL
;  1144     1136  4              THEN
;  1145     1137  4                  IN_MRS = ( .CONV$AB_IN_FAB [ FAB$B_BKS ] * BLOCK_SIZE ) -
;  1146     1138  4                                                                  .IN_VFC - 3
;  1147     1139  4
;  1148     1140  4              ! Indexed
;  1149     1141  4              !
;  1150     1142  4              ELSE
;  1151     1143  4                  IN_MRS = ( .CONV$AB_IN_FAB [ FAB$B_BKS ] * BLOCK_SIZE ) -
;  1152     1144  4                                                                  .IN_VFC - 7
;  1153     1145  4
;  1154     1146  4              END
;  1155     1147  4
;  1156     1148  2          END;
;  1157     1149  2
;  1158     1150  2      ! Now calculate the number of blocks needed.
;  1159     1151  2      !
;  1160     1152  2      ! If UDF, ask for one block extra for overlapping of the buffers
;  1161     1153  2      !
;  1162     1154  2      IF .CONV$AB_IN_FAB [ FAB$B_RFM ] EQLU FAB$C_UDF
;  1163     1155  2      THEN
;  1164     1156  2          OUT_EXTRA = BLOCK_SIZE
;  1165     1157  2      ELSE
;  1166     1158  2          OUT_EXTRA = 0;
;  1167     1159  2
;  1168     1160  3      BEGIN
;  1169     1161  3
;  1170     1162  3      LOCAL
;  1171     1163  3          BYTES,
;  1172     1164  3          VFC_OFFSET;
;  1173     1165  3
;  1174     1166  3      ! Determine which is larger and use that size for the Buffer Size
;  1175     1167  3      !
;  1176     1168  3      BYTES = MAX( BLOCK_SIZE ,                                   ! At least a page
;  1177     1169  3                   ( .IN_MRS + .IN_VFC ),                         ! Input record size
;  1178     1170  3                   ( .CONV$GW_OUT_MRS + .OUT_VFC + .OUT_EXTRA )); ! Output record size
;  1179     1171  3
;  1180     1172  3      ! If we are doing a fast load get some extra bytes to use at the beginning
;  1181     1173  3      ! of the record for control information
;  1182     1174  3      !
;  1183     1175  3      IF .CONV$GL_FAST
;  1184     1176  3      THEN
;  1185     1177  3          BYTES = .BYTES + MAX_REC_CTRL;
;  1186     1178  3
;  1187     1179  3      !
;  1188     1180  3      ! IF UDF input, round buffer up to next whole block
;  1189     1181  3      !
;  1190     1182  3      IF .CONV$AB_IN_FAB [ FAB$B_RFM ] EQLU FAB$C_UDF
;  1191     1183  3      THEN
;  1192     1184  3          BYTES = (.BYTES + 511) AND NOT 511;
;  1193     1185  3
```

```
;  1194          1186  3     ! Create the Buffer from virtural memory
;  1195          1187  !
;  1196          1188        CONV$GL_REC_BUF_PTR = CONV$$GET_VM ( .BYTES );
;  1197          1189
;  1198          1190        ! If we doing a fast load hide the extra bytes at the beginning of
;  1199          1191        ! record.
;  1200          1192        !
;  1201          1193        IF .CONV$GL_FAST
;  1202          1194        THEN
;  1203          1195            CONV$GL_REC_BUF_PTR = .CONV$GL_REC_BUF_PTR + MAX_REC_CTRL;
;  1204          1196
;  1205          1197        ! Set the VFC offset to the max of the two offsets
;  1206          1198        !
;  1207          1199        VFC_OFFSET = MAX( .IN_VFC,.OUT_VFC );
;  1208          1200
;  1209          1201        ! Correct the pointers and set the max. record size
;  1210          1202        !
;  1211          1203        CONV$GL_VFC_BUF_PTR = .CONV$GL_REC_BUF_PTR;
;  1212          1204        CONV$GL_REC_BUF_PTR = .CONV$GL_VFC_BUF_PTR + .VFC_OFFSET;
;  1213          1205
;  1214          1206        CONV$GW_MAX_REC_SIZ = .BYTES - .VFC_OFFSET
;  1215          1207
;  1216          1208        END;
;  1217          1209
;  1218          1210        RETURN CONV$_SUCCESS
;  1219          1211
;  1220          1212    1   END;


                                        0FFC 00000          .ENTRY  CONV$$CREATE_BUFFER, Save R2,R3,R4,R5,R6,-      ; 0959
                                                                    R7,R8,R9,R10,R11
                                59      0000G   CF   9E 00002       MOVAB   CONV$GL_REC_BUF_PTR, R9
                                58      0000G   CF   9E 00007       MOVAB   CONV$AB_IN_FAB+31, R8
                                57      0000G   CF   9E 0000C       MOVAB   CONV$GW_OUT_MRS, R7
                                56      0000G   CF   9E 00011       MOVAB   CONV$AB_OUT_FAB+31, R6
                                03              66   91 00016       CMPB    CONV$AB_OUT_FAB+31, #3                  ; 1022
                                                06   12 00019       BNEQ    1$
                                54        20    A6   9A 0001B       MOVZBL  CONV$AB_OUT_FAB+63, OUT_VFC             ; 1024
                                                02   11 0001F       BRB     2$
                                                54   D4 00021 1$:    CLRL    OUT_VFC                                 ; 1026
                                67        17    A6   B0 00023 2$:    MOVW    CONV$AB_OUT_FAB+54, CONV$GW_OUT_MRS     ; 1032
                                                42   12 00027       BNEQ    6$
                                50        21    A6   D0 00029       MOVL    CONV$AB_OUT_FAB+64, OUT_DEV             ; 1041
                          0D    50              05   E1 0002D       BBC     #5, OUT_DEV, 3$                         ; 1043
                                50        1D    A6   3C 00031       MOVZWL  CONV$AB_OUT_FAB+60, R0                  ; 1045
                                50              54   C2 00035       SUBL2   OUT_VFC, R0
                          67    50              02   A3 00038       SUBW3   #2, R0, CONV$GW_OUT_MRS
                                                2D   11 0003C       BRB     6$
                                51        FE    A6   9A 0003E 3$:    MOVZBL  CONV$AB_OUT_FAB+29, R1                  ; 1049
                                                0D   12 00042       BNEQ    4$
                          08      FF    A6              03   E1 00044       BBC     #3, CONV$AB_OUT_FAB+30, 4$      ; 1050
                          67    01FE    8F              54   A3 00049       SUBW3   OUT_VFC, #510, CONV$GW_OUT_MRS  ; 1052
                                                1A   11 0004F       BRB     6$
                                50        1F    A6   9A 00051 4$:    MOVZBL  CONV$AB_OUT_FAB+62, R0                  ; 1058
```

```
                50              50              09  78 00055          ASHL    #9, R0, R0                                      ;
                                50              54  C2 00059          SUBL2   OUT_VFC, R0                                     ; 1059
                                10              51  91 0005C          CMPB    R1, #16                                         ; 1056
                                                06  12 0005F          BNEQ    5$                                              ;
                67              50              03  A3 00061          SUBW3   #3, R0, CONV$GW_OUT_MRS                         ; 1059
                                                04  11 00065          BRB     6$                                              ; 1058
                67              50              07  A3 00067  5$:     SUBW3   #7, R0, CONV$GW_OUT_MRS                         ; 1065
                                51              6B  9A 0006B  6$:     MOVZBL  CONV$AB_IN_FAB+31, R1                           ; 1072
                                                55  D4 0006E          CLRL    R5                                              ;
                                                51  D5 00070          TSTL    R1                                              ;
                                                2C  12 00072          BNEQ    9$                                              ;
                                                55  D6 00074          INCL    R5                                              ;
                                50      0200    8F  3C 00076          MOVZWL  #512, IN_MRS                                    ; 1076
                                01              66  91 0007B          CMPB    CONV$AB_OUT_FAB+31, #1                          ; 1081
                                                08  12 0007E          BNEQ    7$                                              ;
                0000G   CF      17              A6  B0 00080          MOVW    CONV$AB_OUT_FAB+54, CONV$GW_UDF_MRS             ; 1083
                                                63  11 00086          BRB     15$                                             ;
                0200    8F                      67  B1 00088  7$:     CMPW    CONV$GW_OUT_MRS, #512                           ; 1088
                                                08  1E 0008D          BGEQU   8$                                              ;
                        50 00000000G    8F      D0 0008F          MOVL    #CONV$_UDF_BLK, R0                              ; 1090
                                                04  00096          RET                                                     ;
                0000G   CF      0200    8F      B0 00097  8$:     MOVW    #512, CONV$GW_UDF_MRS                           ; 1092
                                                4B  11 0009E          BRB     15$                                             ; 1081
                                03              51  91 000A0  9$:     CMPB    R1, #3                                          ; 1102
                                                06  12 000A3          BNEQ    10$                                             ;
                                52      20      A8  9A 000A5          MOVZBL  CONV$AB_IN_FAB+63, IN_VFC                       ; 1104
                                                02  11 000A9          BRB     11$                                             ;
                                                52  D4 000AB  10$:     CLRL    IN_VFC                                          ; 1106
                                50      17      A8  3C 000AD  11$:     MOVZWL  CONV$AB_IN_FAB+54, IN_MRS                       ; 1111
                                                38  12 000B1          BNEQ    15$                                             ;
                                51      21      A8  D0 000B3          MOVL    CONV$AB_IN_FAB+64, IN_DEV                       ; 1120
        09                      51              05  E1 000B7          BBC     #5, IN_DEV, 12$                                 ; 1122
                                50      1D      A8  3C 000BB          MOVZWL  CONV$AB_IN_FAB+60, IN_MRS                       ; 1124
                                50              52  C2 000BF          SUBL2   IN_VFC, IN_MRS                                  ;
                                                27  11 000C2          BRB     15$                                             ;
                                53      FE      A8  9A 000C4  12$:     MOVZBL  CONV$AB_IN_FAB+29, R3                           ; 1129
                                                07  12 000C8          BNEQ    13$                                             ;
                                50      0000G   CF  3C 000CA          MOVZWL  CONV$AB_IN_XABFHC+10, IN_MRS                    ; 1131
                                                1A  11 000CF          BRB     15$                                             ;
                                51      1F      A8  9A 000D1  13$:     MOVZBL  CONV$AB_IN_FAB+62, R1                           ; 1137
        51                      51              09  78 000D5          ASHL    #9, R1, R1                                      ;
                                51              52  C2 000D9          SUBL2   IN_VFC, R1                                      ; 1138
                                10              53  91 000DC          CMPB    R3, #16                                         ; 1135
                                                06  12 000DF          BNEQ    14$                                             ;
                                50      FD      A1  9E 000E1          MOVAB   -3(R1), IN_MRS                                  ; 1138
                                                04  11 000E5          BRB     15$                                             ; 1137
                                50      F9      A1  9E 000E7  14$:     MOVAB   -7(R1), IN_MRS                                  ; 1144
                                07              55  E9 000EB  15$:     BLBC    R5, 16$                                         ; 1154
                                53      0200    8F  3C 000EE          MOVZWL  #512, OUT_EXTRA                                 ; 1156
                                                02  11 000F3          BRB     17$                                             ;
                                                53  D4 000F5  16$:     CLRL    OUT_EXTRA                                       ; 1158
                                50              52  C0 000F7  17$:     ADDL2   IN_VFC, R0                                      ; 1169
                                51              67  3C 000FA          MOVZWL  CONV$GW_OUT_MRS, R1                             ; 1170
                                51              54  C0 000FD          ADDL2   OUT_VFC, R1                                     ;
                                51              53  C0 00100          ADDL2   OUT_EXTRA, R1                                   ;
                                51              50  D1 00103          CMPL    R0, R1                                          ;
                                                03  18 00106          BGEQ    18$                                             ;
```

```
                                50              51  D0 00108          MOVL    R1, R0                                          ;
                      00000200  8F              50  D1 0010B 18$:     CMPL    R0, #512                                        ; 1168
                                                05  18 00112          BGEQ    19$                                             ;
                                50      0200    8F  3C 00114          MOVZWL  #512, R0                                        ;
                                51              50  D0 00119 19$:     MOVL    R0, BYTES                                       ;
                                03      0000G   CF  E9 0011C          BLBC    CONV$GL_FAST, 20$                               ; 1175
                                51              0E  C0 00121          ADDL2   #14, BYTES                                      ; 1177
                                0D              55  E9 00124 20$:     BLBC    R5, 21$                                         ; 1182
                                50      01FF    C1  9E 00127          MOVAB   511(R1), R0                                     ; 1184
                          51                    50  CB 0012C          BICL3   #511, R0, BYTES                                 ;
                                50 000001FF     8F  DD 00134 21$:     PUSHL   BYTES                                           ; 1188
                                          0000G 30  00136             BSBW    CONV$$GET_VM                                    ;
                                5E              04  C0 00139          ADDL2   #4, SP                                          ;
                                69              50  D0 0013C          MOVL    R0, CONV$GL_REC_BUF_PTR                         ;
                                03      0000G   CF  E9 0013F          BLBC    CONV$GL_FAST, 22$                               ; 1193
                                69              0E  C0 00144          ADDL2   #14, CONV$GL_REC_BUF_PTR                        ; 1195
                                54              52  D1 00147 22$:     CMPL    R2, OUT_VFC                                     ; 1199
                                                03  18 0014A          BGEQ    23$                                             ;
                                52              54  D0 0014C          MOVL    OUT_VFC, R2                                     ;
                                50              52  D0 0014F 23$:     MOVL    R2, VFC_OFFSET                                  ;
                          0000G CF              69  D0 00152          MOVL    CONV$GL_REC_BUF_PTR, CONV$GL_VFC_BUF_PTR        ; 1203
                                69      0000GDF40   9E 00157          MOVAB   @CONV$GL_VFC_BUF_PTR[VFC_OFFSET], -              ; 1204
                                                                              CONV$GL_REC_BUF_PTR                             ;
                    0000G  CF           51         50  A3 0015D          SUBW3   VFC_OFFSET, BYTES, CONV$GW_MAX_REC_SIZ          ; 1206
                                        50         01  D0 00163          MOVL    #1, R0                                      ; 1210
                                                       04  00166          RET                                                 ; 1212

; Routine Size:  359 bytes,    Routine Base:  _CONV$CODE + 03D0

; 1221          1213  1
; 1222          1214  0 END     ELUDOM


                                                                .EXTRN  LIB$SIGNAL

;                               PSECT SUMMARY
;
;     Name                      Bytes                         Attributes
;
; _CONV$GLOBAL                     8  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,  PIC,ALIGN(2)
; _CONV$CODE                    1335  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(2)


;                               Library Statistics
;
;                                      -------- Symbols --------      Pages       Processing
;     File                             Total    Loaded   Percent      Mapped      Time
;
; _$255$DUA28:[SYSLIB]LIB.L32;1         18619        96         0        1000      00:01.7
; _$255$DUA28:[CONV.SRC]CONVERT.L32;1     165        14         8          17      00:00.2
```

```
;                                       COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:CONVFILES/OBJ=OBJ$:CONVFILES MSRC$:CONVFILES/UPDATE=(ENH$:CONVFILES)

; Size:          1335 code + 8 data bytes
; Run Time:        00:29.5
; Elapsed Time:    01:20.5
; Lines/CPU Min:    2473
; Lexemes/CPU-Min: 18442
; Memory Used:  204 pages
; Compilation Complete
```

CONVDATA
LIS

CONVFSTLD
LIS

CONVFSTIO
LIS

CONVFSTRC
LIS

CONVFILES
LIS

CONVERROR
LIS

CONVFASTM
LIS

CONVDCL
LIS